

SISTEMAS DE RECOMENDACIÓN BASADOS EN TÉCNICAS DE PREDICCIÓN DE ENLACES PARA JUECES EN LÍNEA

MARTA CARO MARTÍNEZ

MÁSTER EN INGENIERÍA INFORMÁTICA – FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin De Máster en Ingeniería Informática

Junio de 2017

Director:

Guillermo Jiménez Díaz

Nota: 10

Autorización de difusión

Marta Caro Martínez

Junio de 2017

El/la abajo firmante, matriculado/a en el Máster en Ingeniería Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “Sistemas de recomendación basados en técnicas de predicción de enlaces para jueces en línea”, realizado durante el curso académico 2016-2017 bajo la dirección de Guillermo Jiménez Díaz en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en internet y garantizar su preservación y acceso a largo plazo.

Resumen

La oferta de todo tipo de productos o experiencias que se puede encontrar en Internet hoy en día es inmensa y difícil de valorar para los usuarios que quieren buscar un producto que se adapte a sus necesidades. Debido a este problema, surgen los sistemas de recomendación, que ayudan a los usuarios a encontrar productos que sean de su interés facilitando sus tareas de búsqueda. Los sistemas de recomendación están implantados en muchísimas plataformas de consumo, pero no en otro tipo de plataformas donde su uso también sería interesante y necesario. Una de estas plataformas son los jueces en línea, donde los sistemas de recomendación podrían ayudar a los usuarios en la selección de los problemas a resolver que les resulten más interesantes.

En este Trabajo Fin de Máster se proponen diferentes métodos de recomendación para implantar en jueces en línea que están basados en grafos de interacciones y que hacen uso de técnicas de predicción de enlaces con el fin de generar recomendaciones. En este trabajo se ha realizado una evaluación de los métodos de recomendación propuestos a través de la generación de experimentos realizados sobre el juez en línea de Acepta el Reto con el objetivo de determinar qué métodos resultan más prometedores.

Palabras clave

Sistemas de recomendación, Jueces en línea, Predicción de enlaces, Grafos de interacción, Acepta el Reto

Abstract

The offer of products or experiences that can be found on the Internet today is immense and difficult to value for users who want to find a product that suits their needs. Because of this problem, recommender systems emerge, which help users to find products that are interesting to them, making easier their search tasks. Recommender systems are implemented in many consumer platforms, but not in other platforms where their use would also be interesting and necessary. One of these platforms are online judges, where recommender systems could help users in selecting which interesting problems to solve.

In this Final Master's Work, different recommender methods to institute in online judges are proposed. They are based on interactions graphs and make use of link prediction techniques in order to generate recommendations. In this work an evaluation of the proposed recommender methods has been made with the generation of experiments on the *Accepta el Reto* (Take On the Challenge) online judge in order to determine which methods are the most promising.

Keywords

Recommender systems, Online Judges, Link prediction, Interactions graphs, Take On the Challenge

Índice general

Índice	I
Índice de figuras	III
Índice de tablas	V
Agradecimientos	VI
1. Introducción	1
1.1. Antecedentes	3
1.2. Objetivos y motivación	4
1.3. Plan de trabajo y organización de la memoria	5
1.4. Introduction	6
2. Sistemas de recomendación	8
2.1. Objetivos de los sistemas de recomendación	9
2.2. Tipos de sistemas de recomendación	13
2.2.1. Sistemas de recomendación basados en filtrado colaborativo	14
2.2.2. Sistemas de recomendación basados en contenido	17
2.2.3. Sistemas de recomendación basados en conocimiento	20
2.2.4. Sistemas de recomendación híbridos	24
2.3. Retos de los sistemas recomendadores y líneas de investigación	28
2.4. Aplicación de los sistemas de recomendación	32
2.5. Conclusiones	34
3. Métodos de recomendación para jueces en línea	36
3.1. Jueces en línea	37
3.2. Técnicas de predicción de enlaces	40
3.3. Recomendación basada en grafos de interacción para jueces en línea	46
3.3.1. Recomendación basada en el grafo de problemas	50
3.3.2. Recomendación basada en el grafo de usuarios	52
3.3.3. Métricas de similitud	53
3.3.4. Métodos de agregación	56
3.4. Conclusiones	59
4. Evaluación y análisis de resultados	60
4.1. Evaluación de los sistemas de recomendación	60
4.1.1. Determinar los objetivos de los usuarios al utilizar el recomendador	61

4.1.2.	Determinar qué conjunto de datos utilizar	64
4.1.3.	Métricas a utilizar durante la evaluación	65
4.1.4.	Proceso de evaluación de los sistemas recomendadores	66
4.2.	Objetivos de la evaluación realizada	68
4.3.	Conjunto de datos: Acepta el Reto	69
4.3.1.	Análisis de Acepta el Reto	71
4.4.	Métricas de evaluación	76
4.5.	Proceso de experimentación	77
4.6.	Análisis de los resultados	80
4.6.1.	Análisis según el grafo de interacciones	81
4.6.2.	Análisis según el método de agregación	82
4.6.3.	Variaciones sobre el número de elementos recomendados	87
4.7.	Conclusiones	92
5.	Conclusiones	94
5.1.	Conclusions	97
5.2.	Líneas de trabajo futuro	100
	Bibliografía	102

Índice de figuras

2.1.	Proceso de recomendación de un recomendador basado en casos [2].	23
2.2.	Proceso de recomendación de un recomendador basado en restricciones [2]. .	23
2.3.	Diseño paralelo de un recomendador híbrido [2].	26
2.4.	Diseño secuencial de un recomendador híbrido [2].	26
2.5.	Clasificación de las técnicas de para combinar estrategias de recomendación en los tres tipos de diseños híbridos [2].	28
3.1.	Ejemplo de un problema de predicción de enlaces [50]	42
3.2.	Diferencia de procesamiento entre los diferentes tipos de técnicas de predicción de enlaces [54]	42
3.3.	Transformación de un grafo bipartito en uno no bipartito a través del método <i>bipartite network projection</i> [58].	49
3.4.	Generación de una recomendación basada en el grafo de problemas.	51
3.5.	Generación de una recomendación basada en el grafo de usuarios.	52
3.6.	Ejemplo de uso de los sistemas de votación	58
4.1.	Gráfico de envíos en Acepta el Reto entre febrero de 2014 y febrero de 2017, categorizados por veredicto.	74
4.2.	Gráfico acumulado de envíos en Acepta el Reto entre febrero de 2014 y febrero de 2017, categorizados por veredicto.	75
4.3.	Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación <i>Precision</i> con el método de agregación basado en la similitud máxima	87
4.4.	Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación <i>One hit</i> con el método de agregación basado en la similitud máxima	88
4.5.	Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación MRR con el método de agregación basado en la similitud máxima	88
4.6.	Valores obtenidos para el grafo de problemas, $t = 20/10/2016$ y la métrica de evaluación <i>Precision</i> con el método de agregación basado en la similitud máxima	89
4.7.	Valores obtenidos para el grafo de problemas, $t = 20/10/2016$ y la métrica de evaluación <i>One hit</i> con el método de agregación basado en la similitud máxima	89
4.8.	Valores obtenidos para el grafo de problemas, $t = 20/10/2016$ y la métrica de evaluación MRR con el método de agregación basado en la similitud máxima	90
4.9.	Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación <i>Precision</i> con los sistemas de votación como método de agregación y CN como métrica de similitud	90

4.10. Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación <i>One hit</i> con los sistemas de votación como método de agregación y CN como métrica de similitud	91
4.11. Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación MRR con los sistemas de votación como método de agregación y CN como métrica de similitud	91

Índice de tablas

2.1.	Diferencias entre los principales tipos de sistemas de recomendación [2] . . .	21
4.1.	Análisis de la base de datos de Acepta el Reto.	72
4.2.	Resultados de la evaluación con las diferentes métricas de similitud con base en el grafo de usuarios y de problemas para listas de recomendación de $k = 5$ y $t = 20/10/2016$. Los mejores resultados obtenidos entre el grafo de usuarios y el de problemas está marcado en negrita. El mejor valor obtenido por métricas está marcados con *.	82
4.3.	Resultados de la evaluación del sistema basado en el grafo de usuarios: se muestran los resultados obtenidos con y sin sistemas de votación para $k = 5$ y $t = 20/10/2016$. Los mejores resultados obtenidos entre el grafo de usuarios y el de problemas está marcado en negrita. El mejor valor obtenido por métricas está marcados con *.	84
4.4.	Resultados de la evaluación del sistema basado en el grafo de problemas: se muestran los resultados obtenidos con y sin sistemas de votación para $k = 5$ y $t = 20/10/2016$. Los mejores resultados obtenidos entre el grafo de usuarios y el de problemas está marcado en negrita. El mejor valor obtenido por métricas está marcados con *.	86

Agradecimientos

Quería dar las gracias a todas aquellas personas que me han ayudado a que este Trabajo Fin de Máster se lleve a cabo.

Principalmente quería agradecer a Guille el esfuerzo y dedicación que ha puesto de nuevo en mi trabajo, por la ayuda y los consejos que me ha dado durante los últimos tres años y por hacerme descubrir que quiero dedicarme a la investigación. Quería agradecer también al grupo GAIA la oportunidad que me ha brindado de trabajar en el proyecto y permitirme ir a presentar mi TFM al ICCBR de este año.

Agradecer a mis compañeros y amigos del Máster en Ingeniería Informática los buenos momentos vividos. En especial, a Fer, por ser el mejor amigo y compañero de prácticas que podría haber tenido, a mis amigos del aula 16, por el apoyo y la ayuda diaria pero sobre todo por hacer que cada día sea aún más divertido que el anterior, y a Mariela, por todas las veces que me ha obligado a desconectar de este TFM para ir a merendar.

Por último, quiero agradecer al resto de mis amigos y a mi familia el apoyo incondicional que me han demostrado. Sobre todo a mi abuela Marta, a mi hermana Lidia y a mis padres por la paciencia, consejos y cariño que me han dado durante los buenos y malos momentos que he pasado durante la realización del máster. A mi abuelo Luis, por lo anterior y por la enorme fuerza que me transmite.

Este Trabajo Fin de Máster ha sido realizado en el marco del proyecto de investigación TIN2014-55006-R financiado por el MINECO.

*A mis abuelos Luis y Marta,
como todo lo que hago,
también esto es para vosotros*

Capítulo 1

Introducción

Los sistemas de recomendación son los sistemas software que permiten a un usuario obtener propuestas de productos en base a sus gustos. Surgen de la necesidad de las personas de determinar qué productos son los que más le pueden interesar para una determinada actividad de entre todas las posibles opciones que hoy en día tenemos a nuestro alcance. La aparición de la Web y de las plataformas de venta y consumo en Internet han propiciado que la oferta de productos y actividades a las que podemos acceder aumente a pasos agigantados, siendo imposible el análisis de toda la oferta de productos. Con los sistemas de recomendación, se facilita considerablemente esta tarea y se proporciona al usuario una herramienta para conseguir lo que busca en menos tiempo y de forma mucho más efectiva.

Existen diferentes tipos de sistemas de recomendación, los cuales se pueden aplicar a diferentes tipos de plataformas, de muchos ámbitos distintos. La mayor parte de los sistemas de recomendación ya implantados y altamente desarrollados se encuentran en el ámbito del comercio electrónico (como en Amazon o eBay) y en plataformas de consumo de entretenimiento: música (Spotify), series y películas (Netflix), etc. Sin embargo, también existen otros ámbitos en los que hay plataformas donde sería necesario implantar un sistema de recomendación para ayudar a los usuarios en las tareas que realizan en la misma. Un ejemplo sería el campo de la enseñanza o el aprendizaje, donde la existencia de sistemas de recomendación

no está tan extendido. Es cierto que ya existen muchas plataformas web en este ámbito que tienen su propio sistema de recomendación, como en Coursera, donde se puede acceder a MOOCs (*Massive Online Open Courses*, en español Cursos Online Masivos y Abiertos). Sin embargo, en otros muchos sistemas no existen, como en el caso de los jueces en línea. Los jueces en línea son plataformas que proporcionan multitud de problemas a estudiantes de programación y ciencias de la computación, para que puedan realizar ejercicios y mejorar sus capacidades. También se utilizan como medios para celebrar concursos de programación. A pesar de su creciente uso y la clara utilidad de los jueces en línea en el ámbito educativo, estos sistemas no cuentan con sistemas de recomendación que ayuden a sus usuarios a encontrar nuevos problemas que puedan interesarles.

Los sistemas de recomendación funcionan en base a las valoraciones que los usuarios realizan sobre los productos de una plataforma. También pueden trabajar en función de las descripciones de los productos o del perfil del usuario. Sin embargo, existen plataformas de las que no podemos conseguir esta información porque los usuarios no realizan valoraciones en ella o porque no existen descripciones de los productos. En estos casos, la única información disponible son las interacciones que los usuarios realizan con los productos de la plataforma (el usuario ha visto una película, ha resuelto un problema, etc.). Como consecuencia, surgen los sistemas de recomendación basados en grafos, los cuales generan las recomendaciones en base a un grafo que representa las interacciones realizadas en la plataforma.

Este Trabajo Fin de Máster es un trabajo de investigación en el que se proponen varios modelos de recomendación basados en la utilización de grafos de interacción y en técnicas de predicción de enlaces (o *link prediction*) sobre esos grafos. Más concretamente, se ha propuesto un modelo de recomendación aplicable sobre los jueces en línea, que normalmente carecen de sistemas de recomendación. La propuesta se ha completado con una evaluación

de los métodos de recomendación utilizando el juez en línea Acepta el Reto (el juez en línea de la Facultad de Informática de la Universidad Complutense de Madrid) como plataforma sobre la que se va a evaluar el funcionamiento y la efectividad de los algoritmos desarrollados. Se han analizado también los resultados obtenidos en esta evaluación con el fin de determinar cuáles son las técnicas que ofrecen mejores resultados.

A continuación se van a detallar algunos aspectos de este trabajo que es necesario conocer como introducción en esta memoria. Así, se va a determinar cuáles son los antecedentes de este Trabajo Fin de Máster, cuáles son los objetivos para proponer estos modelos de recomendación y cuál ha sido el plan de trabajo realizado.

1.1. Antecedentes

Los sistemas de recomendación son sistemas implantados desde hace tiempo en páginas web y plataformas de consumo de productos. Desde la aparición de los primeros artículos de investigación sobre sistemas de recomendación a mediados de la década de los noventa [36], la investigación en este campo ha aumentado considerablemente, aunque aún existen diferentes tendencias sobre sistemas de recomendación en las que hay que trabajar en el futuro. Por tanto, los antecedentes sobre sistemas de recomendación son muchos y variados.

El antecedente de este trabajo en concreto es el trabajo realizado el año pasado por profesores de la Facultad de Informática (UCM) y miembros del grupo de investigación GAIA, en el que se enmarca este Trabajo Fin de Máster. Los trabajos realizados están publicados en el artículo de Jimenez-Diaz et al. *"Similarity Metrics from Social Network Analysis for Content Recommender Systems"* [21], del congreso *International Conference on Case-Based Reasoning* (ICCBR) de 2016. Se trata de la propuesta de un método de recomendación para jueces en línea a partir de un grafo de interacción de problemas contenidos en el juez. Se ha

evaluado el método con el juez en línea Acepta el Reto.

En este Trabajo de Fin de Máster se ha reproducido el trabajo de Jimenez-Diaz et al. y se ha ampliado la investigación en este campo. Junto con el estudio y réplica de esta publicación, se incluyen nuevas propuestas de recomendadores basadas en el grafo de interacción de los usuarios de la plataforma, así como la inclusión de nuevos métodos que cambian la forma de recomendar problemas a los usuarios. El proceso de evaluación ha sido desarrollado tanto con el trabajo realizado el año pasado como con los nuevos sistemas implementados, y se ha realizado un nuevo análisis global con los datos obtenidos. La etapa de evaluación ha dado lugar a una inmensa cantidad de datos que ha sido necesario analizar desde cero, para así poder comparar los resultados generados en este TFM.

1.2. Objetivos y motivación

El objetivo principal de este trabajo es proponer diferentes métodos de recomendación para jueces en línea basados en grafos de interacciones y en técnicas de predicción de enlaces.

El conjunto de objetivos secundarios de este TFM en base al principal es:

1. Investigar sobre sistemas de recomendación: qué son, cómo funcionan, qué tipos existen, qué usos tienen, por qué los usuarios lo encuentran necesario y útil y cómo se evalúan.
2. Ampliar los conocimientos básicos sobre grafos y Análisis de Redes Sociales, teniendo en cuenta las técnicas que pueden resultar interesantes para este trabajo.
3. Estudiar en qué consisten las técnicas de predicción de enlaces e investigar cómo se aplican en los sistemas de recomendación.

4. Construir el modelo de sistema de recomendación desarrollado en el trabajo de 2016: entender cómo funciona este modelo, comprender por qué se aplican las técnicas utilizadas y replicar la implementación del modelo como forma de afianzar la base del trabajo para su ampliación.
5. Diseñar los nuevos modelos de sistema de recomendación basados en grafos de interacción, aplicando técnicas de predicción de enlaces sobre esos grafos.
6. Evaluar los métodos de recomendación propuestos usando un conjunto de datos extraído del juez en línea Acepta el Reto.

1.3. Plan de trabajo y organización de la memoria

La metodología seguida en este trabajo ha sido una metodología investigadora: se comienza definiendo el problema de investigación (la falta de orientación a los usuarios de jueces en línea), después se hace una propuesta de solución (la implantación de sistemas de recomendación), luego se elige la metodología para solucionar el problema (utilización de grafos basados en interacciones y técnicas de predicción de enlaces para proponer recomendadores) y, por último, se evalúan las propuestas de solución al problema (evaluación de los modelos recomendadores), analizando los resultados y la validez de las propuestas.

El plan de trabajo inicial se ha basado en la aplicación de esta metodología para cubrir los objetivos contemplados en la anterior sección. Primero se ha realizado un estado del arte sobre sistemas de recomendación, detallando qué son estos sistemas, sus tipos, la motivación de su existencia y las líneas de investigación abiertas actualmente en este campo (Capítulo 2). Después, se ha hecho un repaso sobre la existencia de los jueces en línea y su estado actual, así como sobre la necesidad de incluir sistemas de recomendación en estas plataformas. También se ha estudiado qué son las técnicas de predicción de enlaces

y cómo se aplican. Después, se ha comenzado la réplica del trabajo realizado el año pasado y tras esto, se ha hecho una propuesta de un conjunto de métodos de recomendación basados en los algoritmos de predicción de enlaces (Capítulo 3). Luego se ha iniciado el proceso de evaluación de los métodos implementados, haciendo un repaso sobre cómo se hace una evaluación sobre un sistema de recomendación y aplicando estos conocimientos sobre la experimentación realizada con nuestros modelos de recomendación. Tras esto, se ha realizado un análisis de los datos obtenidos en la evaluación (Capítulo 4). Por último, se han detallado las conclusiones tras el trabajo realizado y las líneas de trabajo futuro (Capítulo 5).

1.4. Introduction

Recommender systems are software systems that allow a user to get products based on their tastes. The origin of recommender systems is in the the need of people to determine which products are interesting for them. The Web birth and the sales on the Internet have made possible to access to a huge offer of products: analyzing this offer is impossible. With recommender systems, this task is greatly facilitated. The user is provided with a tool to get what he or she is looking for in less time and in a much more effective way.

There are different types of recommendation systems, which can be applied to different types of platforms. Most of the already developed recommender systems are in the field of electronic commerce (as in Amazon or eBay) and in entertainment platforms: music (Spotify), series and movies (Netflix), etc. However, there are also other areas in which there are platforms where it would be necessary to implement a recommendation system. An example would be the teaching or learning field, where the existence of recommender systems is not widespread. It is true that there are already many web platforms in this area that have their own recommendation system, as in Coursera. However, recommender systems do not exist in many other systems, as in the case of online judges. Online judges are platforms

that provide problems for programming and computer science students. They are also used in programming contests. Despite their increasing use and the utility of online judges in education, these systems do not have recommender systems that help their users find new problems that may interest them.

The recommender systems work based on the ranking that the users make on the platform. They can also work with the descriptions of the products or the user profiles. However, there are platforms from which we can not get this information because the users do not make rankings in it or because there are not descriptions of products. In these cases, the information available is the interactions that users make with the products of the platform (the user has seen a movie, the user has solved a problem, etc.). As a consequence, graph-based recommendation systems emerge, which generate recommendations based on a graph that represents the interactions performed on the platform.

This Final Master's Work is a research work that proposes several recommender models based on the use of interaction graphs and link prediction techniques on these graphs. Specifically, a recommender model has been proposed for online judges, which normally lack recommender systems. The proposal has been completed with an evaluation of our recommender methods using the online judge Acepta el Reto or Take On the Challenge (the Complutense University of Madrid (UCM) online judge) as a platform on which to evaluate the effectiveness of the developed algorithms. We have also analyzed the results obtained in this evaluation in order to determine which are the techniques that offer better results.

Capítulo 2

Sistemas de recomendación

Actualmente, vivimos en una sociedad en la que continuamente recibimos y creamos información gracias a las nuevas tecnologías: estamos inmersos en la Era Digital o de la Información. Es tan elevada la información que nos llega que se hace difícil discernir qué es lo que de verdad nos interesa y lo que no. Lo mismo ocurre cuando queremos adquirir un producto (comprar un producto, ver una película, leer un libro o escoger un lugar de vacaciones): entre tantas opciones que se pueden encontrar gracias a las nuevas tecnologías, resulta complicado clasificar la información y decidir cuál es el producto que mejor se adapta a nuestras necesidades.

Con la aparición de la Web y su posterior desarrollo, los sistemas de recomendación surgen como forma de que los usuarios puedan conocer qué productos son los que más les interesan dentro de la gran variedad entre la que pueden escoger. Por tanto, estos sistemas ayudan a encontrar productos que cubran sus necesidades y se encuentren dentro de sus preferencias, haciendo más fácil la tarea de clasificar y adquirir productos a través de las nuevas tecnologías.

En este capítulo se hará una revisión de qué son los sistemas de recomendación, qué tipos de recomendadores existen y cómo funciona cada uno de ellos, cuáles son los retos futuros en investigación en este campo y qué ejemplos de sistemas de recomendación destacables

existen actualmente.

2.1. Objetivos de los sistemas de recomendación

La Web, como medio de transacciones sociales o empresariales, ha ido aumentando en importancia gracias entre otras cosas, a la creación y desarrollo de plataformas como pueden ser las tiendas online o medios de intercambio de información, como las redes sociales. Estas plataformas permiten a sus usuarios opinar sobre productos que le interesan. Por ejemplo, los usuarios de Netflix¹, página web donde sus usuarios pueden ver contenidos en streaming, como películas o series, pueden valorar estos contenidos en función de sus gustos. Por otro lado, existen otras plataformas en las que los usuarios no emiten opiniones explícitas de los contenidos o productos ofrecidos en la plataforma, pero que pueden ser incluso más efectivas. En cualquier tienda online, como Amazon² o eBay³, se recopilan datos de los gustos de sus usuarios en función de las compras que realizan en la propia web. Toda esta información es realmente relevante para poder aplicar sistemas de recomendación en estas plataformas, donde la necesidad de incluir este tipo de sistemas se hace patente cuando los usuarios necesitan explorar el catálogo de productos de Amazon, o el catálogo de series de Netflix, y encontrar aquello que les sea más útil o interesante, tardando el menor tiempo posible durante la búsqueda de los productos [2].

Un sistema recomendador se puede definir como un sistema que ayuda a asociar de manera personalizada un producto con un usuario [20]. La función que aplican los algoritmos de recomendación asigna un valor real al interés de un usuario U por un producto I : $f : U \times I \rightarrow R$ [55]. El principio básico de los algoritmos de recomendación, por tanto, es encontrar las dependencias entre la actividad de los usuarios y la actividad que se realiza

¹<https://www.netflix.com/es/>

²<https://www.amazon.es/>

³<http://www.ebay.es/>

sobre los productos. Si, por ejemplo, un usuario está interesado en una película de acción, es más probable que esté interesado en otra película de acción o de aventuras, que, por ejemplo, en un documental sobre naturaleza. Es decir, existen correlaciones entre las diferentes categorías de productos, lo que va a hacer que se puedan hacer recomendaciones más efectivas al conocer que varios productos de diferentes categorías son interesantes al usuario por existir esta correlación. Las dependencias en las que se basan los algoritmos de recomendación pueden estar fundamentadas en este tipo de correlaciones, pero también pueden ser deducidas a partir de características individuales de cada producto. Se pueden extraer características comunes de los productos, por ejemplo a partir de las calificaciones dadas a los productos, y utilizar esta información para proporcionar las recomendaciones. Existen muchos mecanismos de aprendizaje para deducir y extraer estas características, por ejemplo, para hacer agrupaciones de productos según los usuarios con los que han interactuado [2].

Teniendo en cuenta las necesidades de los usuarios a la hora de navegar por la web, los objetivos de los sistemas de recomendación pueden ser, de manera general, predecir las valoraciones que realizan los usuarios con los productos con los que interactuarán o predecir cuáles son los productos con los que interactuarán los usuarios [2].

Predecir las valoraciones que realizan los usuarios sobre los productos consiste en conseguir determinar qué calificación o valoración pondría un usuario a un determinado producto en función de sus intereses, gustos o satisfacción con el producto. Las interacciones ocurridas en la plataforma se pueden representar a través de una matriz A de dimensiones $m \times n$, donde m es el número de usuarios y n , el número de productos. En la posición A_{ij} se establecerá la valoración que el usuario i ha adjudicado al producto j . Uno de los objetivos de los algoritmos de recomendación es, por tanto, completar esta matriz, prediciendo las valoraciones de los usuarios para los productos que aún no han valorado.

Por otro lado, predecir cuáles son productos con los que interactuará cada usuario consiste en encontrar los k mejores productos para un usuario, o encontrar los k usuarios a los que podría interesarles un producto. En general, es más habitual considerar los k mejores productos para un usuario.

Según Schafer et al. [45], las principales tareas que los sistemas de recomendación pueden realizar y por las que los usuarios los utilizan son:

- **Ayudar a los usuarios a encontrar productos o usuarios que puedan interesarles.** Como se ha comentado al inicio del capítulo, los sistemas de recomendación nos hacen la vida más fácil ayudándonos a elegir entre la amplia oferta que nos proporciona la Web. Es importante mencionar que este tipo de ayuda también se puede aplicar a grupos de personas, cuando es necesario cubrir todas las necesidades de un grupo de personas que buscan un producto. Por ejemplo, cuando un grupo de amigos quiere ir al cine a ver una película, el sistema tiene que cumplir los requisitos y restricciones de los usuarios adaptándolos de la mejor manera posible.
- **Aconsejar al usuario sobre un determinado producto.** Si un usuario está interesado en un producto concreto, a través de los sistemas recomendadores, se puede conocer la opinión de la comunidad con respecto a ese producto.
- **Ayudar a encontrar una mezcla de productos nuevos y viejos.** Ayudar a encontrar productos nuevos que nos puedan interesar, pero también productos con los que ya hayamos interactuado porque podemos volver a necesitarlo. Por ejemplo, al hacer la lista de la compra podemos volver a comprar alimentos que ya hayamos comprado anteriormente o podemos necesitar comprar productos nuevos para llegar a tener una dieta equilibrada.
- **Ayudar a encontrar productos según nuestra situación o nuestras necesidades.** Es importante que el sistema recomendador nos ofrezca unos productos

determinados en la situación en la que lo necesitemos. No es lo mismo necesitar la recomendación de una película para ir a verla con amigos, que para ir a verla con familia. Tampoco es lo mismo necesitar la recomendación de un restaurante para una primera cita que para ir con amigos. De ahí la necesidad de adaptar los sistemas recomendadores para obtener los mejores resultados posibles.

Teniendo en cuenta las tareas que los usuarios pretenden realizar con el recomendador, se pueden definir las principales funcionalidades de los sistemas de recomendación, de forma que se cubran las necesidades que tienen los usuarios y se cubran también las funcionalidades que los usuarios esperan encontrar en este tipo de sistemas. A continuación se detallan estas funcionalidades [45]:

- **Recomendar productos.** El sistemas recomendador puede mostrar una lista de productos al usuario que puedan interesarle y mostrarlos además ordenados por los intereses del usuario.
- **Predecir las valoraciones de un determinado producto.** Se deduce cuáles pueden ser las valoraciones para un producto específico.
- **Hacer recomendaciones a partir de un conjunto de datos determinado.** El sistema se tiene que encargar de realizar las recomendaciones teniendo en cuenta las restricciones de los propios usuarios. De esta forma se tiene que hacer un filtro del conjunto de datos original, y así generar las recomendaciones sobre el conjunto de datos filtrado. El ejemplo que se muestra en el trabajo de Schfer et al. [45] es la recomendación de películas para un niño o niña menor de ocho o nueve años años, quien no quiere ver películas que no sean recomendadas para menores de dieciocho años ni películas violentas.

2.2. Tipos de sistemas de recomendación

La variedad de algoritmos que pueden utilizar los sistemas de recomendación es amplia, lo que da lugar a que se formen diferentes familias o tipos de sistemas recomendadores. Los sistemas de recomendación se pueden clasificar según las siguientes características [39]:

- **Quién lleva la iniciativa en el proceso de recomendación.** En base a esto, el sistema recomendador puede ser de dos tipos: *reactivo*, cuando es el usuario quien busca un nuevo producto, y el sistema le recomienda en base a ello, o *proactivo*, si el sistema recomienda productos basándose en búsquedas, compras o valoraciones anteriores del usuario, siendo el sistema el que recomienda al usuario directamente sin que el usuario se lo pida [32].
- **La dinámica de la recomendación** [48]. Por un lado, existen los sistemas de recomendación *single-shot*, con los que el usuario del sistema puede elegir uno de los productos de entre los recomendados, o descartarlos todos. Si quiere volver a encontrar nuevas recomendaciones, tendrá que hacer una nueva búsqueda. Por otro lado, existen los sistemas de recomendación *conversacionales*, con los que se realiza un proceso iterativo en el que el usuario puede ir refinando los requisitos de lo que busca hasta encontrar lo idóneo. A su vez, los conversacionales se pueden dividir en otros dos tipos [47]: los de *navegación-por-propuesta*, con los que el usuario puede refinar sus requisitos a partir de un conjunto de productos mostrado, y los de *navegación-por-pregunta*, con los que el usuario responde a un conjunto de preguntas, de forma que el sistema pueda recoger los datos de los requisitos determinados por el usuario.
- **Determinación de la calidad de los elementos.** Ciertos recomendadores buscan la calidad en las recomendaciones que realizan, es decir, que buscan encontrar elementos con una cierta diversidad a los demás productos recomendados. De esta forma, se asegura encontrar elementos que cumplan con los requisitos de la consulta pero que sean lo más diferentes posible entre sí para ofrecer diversidad al usuario [31].

Independientemente de lo anterior, los sistemas de recomendación utilizan técnicas basadas en la información que pueden obtener del conocimiento de los productos, de las valoraciones de los usuarios, de la interacción del usuario con el recomendador o de la interacción del usuario con los productos. En base a esto, se realiza una clasificación en diferentes tipos de recomendadores [2] (esta es la clasificación más extendida):

- Los sistemas de recomendación basados en **filtrado colaborativo**. Se basan en las valoraciones que han dado los usuarios sobre los productos.
- Los sistemas de recomendación basados en **contenido**. Son los que tienen en cuenta la descripción de los productos para hacer las recomendaciones.
- Los sistemas de recomendación basados en **conocimiento**. Son aquellos que utilizan todo el conocimiento que se puede obtener de los usuarios y productos.
- Los sistemas de recomendación **híbridos**. Se trata de sistemas que combinan técnicas de los anteriores recomendadores para intentar cubrir sus deficiencias.

A continuación se detallarán cómo funcionan cada uno de los anteriores tipos de sistemas recomendadores, comentando cuáles son las principales características y debilidades de cada uno de ellos.

2.2.1. Sistemas de recomendación basados en filtrado colaborativo

Los sistemas de recomendación basados en filtrado colaborativo son los recomendadores más usados de entre todos los mencionados. El filtrado colaborativo es el proceso por el cual se calculan o estiman las valoraciones de productos usando la opinión de diferentes personas [45]. Empareja usuarios con gustos parecidos, o productos con valoraciones similares [1, 12]. Por tanto, estos recomendadores se basan en las valoraciones que otros

usuarios han realizado sobre los productos o en las valoraciones que ha hecho un determinado usuario sobre determinados productos parecidos. Se puede concluir que este tipo de sistemas no necesitan información sobre el producto en sí para hacer las recomendaciones, sino que lo que utiliza son las valoraciones realizadas por los usuarios del sistema. Es por esto por lo que, en general, son sistemas que tienen una complejidad de cálculo muy alta [41].

Para implementar este tipo de recomendadores se pueden usar varias técnicas [22]: las estrategias basadas en memoria y las estrategias basadas en modelos.

- **Estrategias basadas en memoria**, a las que también se las llama estrategias basadas en vecinos. Son las que hacen recomendaciones basándose en las valoraciones realizadas por los usuarios del sistema [41]. Han sido las estrategias más ampliamente usadas en la implementación de sistemas recomendadores de filtrado colaborativo. Dentro de estas técnicas, existen dos subconjuntos [22]:
 - **Basado en usuario**: se predice la valoración de un usuario para un producto, teniendo en cuenta las valoraciones que han hecho los vecinos del usuario (aquellos usuarios que han valorado de forma parecida los mismos productos que él) para ese mismo producto.
 - **Basado en producto**: se predice la valoración de un usuario para un producto, teniendo en cuenta las valoraciones que ha hecho ese usuario para vecinos del producto (aquellos productos que han sido valorados de forma parecida por los usuarios).
- **Estrategias basadas en modelos**. El problema de las estrategias basadas en vecinos es que la complejidad de cálculo es enorme. Es por esto por lo que surgieron las estrategias basadas en modelos, con las que se construyen modelos estadísticos de patrones de valoración de usuarios y productos para hacer predicciones automáticas de valoraciones. Con ellas se realiza un preprocesamiento *offline* de los datos de la

plataforma para hacer las recomendaciones y predicciones con esos datos [8]. Entre ellos se encuentran los modelos de semántica latente [18], redes neuronales [43] o probabilísticos [52, 53], pero el que más destaca es el modelo de los factores latentes [23]. Es el método más utilizado por la mejora de precisión que ha proporcionado gracias a la utilización de la técnica de factorización de matrices. Lo que hace esta técnica es categorizar usuarios o productos en unas determinadas clases deducidas de las valoraciones aportadas por los usuarios [41].

Aunque los algoritmos de filtrado colaborativo sean el tipo de sistema recomendador más utilizado y popular por la eficacia de sus recomendaciones aún tienen debilidades que provocan que las recomendaciones no sean totalmente efectivas y que, por lo tanto, es necesario que se solventen [41]. Por un lado, tenemos dispersión: el conjunto de valoraciones usuario/producto normalmente es menor que el total de valoraciones posibles. Esto dificulta encontrar valoraciones realizadas por el vecino del usuario y sobre el vecino del producto.

También tenemos el problema de la existencia de usuarios poco habituales. Es difícil hacer recomendaciones a usuarios cuyas valoraciones no sean habituales con respecto a las de los demás usuarios. No se conocen los gustos de una persona por actividades puntuales en la plataforma, por lo que tampoco se pueden realizar recomendaciones fiables en función de una única o unas pocas interacciones del usuario con los productos de la plataforma.

El problema que se produce con los nuevos productos y los nuevos usuarios es el mismo que el problema de los usuarios poco habituales. Hasta que el producto no haya recibido una cierta cantidad de valoraciones, no se podrá establecer la recomendación.

Por último, tenemos el problema de la fiabilidad. Las recomendaciones pueden alterarse si una empresa empieza a hacer valoraciones positivas de sus productos o negativas de

productos de la competencia. Es necesario, por tanto, implementar sistemas robustos que impidan este tipo de comportamientos.

A pesar de esto, es uno de los mejores sistemas para que los usuarios consigan recomendaciones personalizadas, además de evitarse el tener que evaluar todos los productos y tener que filtrar la información por ellos mismos. Esto, por otro lado, tiene como desventaja que el usuario siempre tiene que hacer alguna valoración en la plataforma, aunque no tenga una opinión formada sobre ningún producto. Esto no ocurre con otro tipo de sistemas de recomendación.

2.2.2. Sistemas de recomendación basados en contenido

Existen otro tipo de sistemas de recomendación que no utilizan los patrones de valoración de los usuarios del sistema para hacer recomendaciones. Uno de estos tipos de sistemas son los sistemas de recomendación basados en contenido [2, 41]. Se basan principalmente en la descripción de los productos de la plataforma y la información de perfil de los usuarios para generar las recomendaciones. Un producto se representa por su descripción o por palabras importantes relacionadas con ese producto. A partir de ello, se buscan coincidencias entre esa representación y el perfil del usuario. Es por esto por lo que este tipo de sistemas son muy eficaces en el caso de la recomendación de productos nuevos, ya que se basan en las características del producto y no en las inexistentes valoraciones de ese producto. En muchos casos, la información son atributos tipo palabras clave que son extraídos de la descripción de los productos. Teniendo todo esto en cuenta, es fácil concluir que los sistemas de recomendación basados en contenido son muy usados en contextos donde hay una gran cantidad de información disponible sobre los productos. Son muy adecuados en contextos de gran riqueza textual y dominios no estructurados, como por ejemplo las páginas web.

Para generar el perfil del usuario, estos sistemas utilizan la información de los usuarios extraída de sus interacciones con los productos de la plataforma [28, 37]. Por tanto, los datos extraídos del resto de usuarios no se van a utilizar para generar las recomendaciones al usuario objetivo, lo cual tiene ventajas y desventajas, dependiendo del escenario en el que queramos aplicar los sistemas. Por un lado, supone una ventaja cuando existen pocas valoraciones de los usuarios para un determinado producto. Al no necesitarse las valoraciones del resto de usuarios, tan sólo la información de perfil del usuario objetivo, resulta más sencillo hacer las recomendaciones en este tipo de escenarios. Como se ha comentado anteriormente, ocurre lo mismo cuando aparece un nuevo producto en la plataforma. Al no existir valoraciones sobre ese producto, resultará más sencillo recomendarlo a los usuarios a través de un sistema recomendador basado en contenido, ya que, para ello, sólo se utilizará su descripción. Por otro lado, supone una desventaja cuando aparecen nuevos usuarios en el sistema. Al no usar las valoraciones del resto de usuarios, va a haber muy poca diversidad en las recomendaciones que se le hagan al usuario objetivo, además de que es muy probable que esas recomendaciones sean productos con los que el usuario ya haya interactuado.

El origen de los datos que utilizan los sistemas de recomendación de contenidos provienen de dos fuentes distintas [2]:

- La primera fuente de datos son los **atributos de los productos** de la plataforma, obtenidos de sus propias descripciones. Por ejemplo, se pueden obtener estos atributos a partir de las etiquetas del propio producto o de la descripción que incluya el administrador al introducir el producto en la base de datos del sistema.
- La segunda fuente de datos sería el **perfil del usuario**. La información del perfil de usuario se genera a partir de las interacciones que el usuario realiza con los productos de la plataforma. Se utiliza esta información como forma de relacionar al usuario con productos con los que haya interactuado y que tengan características que le interesen. No se relaciona al usuario con productos con los que ya ha interactuado. El *feedback*

que proporciona el usuario puede ser explícito, cuando sea información obtenida de las valoraciones que ha realizado, o implícito, si la información se extrae de las acciones que ha realizado el usuario.

Es importante usar datos estructurados o estandarización de descripciones en este tipo de sistemas para poder manejar tanta variedad de productos, modelos de descripciones y perfiles de usuarios. En muchos casos, si no se encuentra ya en ese formato, se convierte la información en palabras clave con el objetivo de trabajar mejor con el contenido. A partir de esto, se pueden determinar los diferentes pasos que realizan estos sistemas para trabajar [2].

1. **Preprocesamiento y extracción de características.** Se extrae la información básica de todos estos sistemas para hacer una representación en palabras clave de toda la información que describe a los productos. Esta etapa es muy importante para maximizar la efectividad de los sistemas basados en contenido.
2. **Aprendizaje basado en el perfil de usuario.** Se trata de generar un modelo específico para cada usuario para predecir sus intereses en los diferentes productos teniendo en cuenta su actividad (*feedback* implícito) y las valoraciones realizadas (*feedback* explícito). Este modelo de aprendizaje personalizado para cada usuario objetivo se construye a partir de un conjunto de entrenamiento de forma muy similar a como se usa en modelos de clasificación o en modelos de regresión.
3. **Filtrado y recomendación.** El modelo de aprendizaje construido en la etapa anterior se va a utilizar para generar las recomendaciones al usuario objetivo.

Como ocurre con los algoritmos de filtrado colaborativo, también los sistemas de recomendación basados en contenido tienen debilidades [41]. En el conjunto de estas debilidades destaca la sobreespecialización. Se produce porque se recomiendan productos en función de

intereses anteriores, y no se tienen en cuenta las variaciones en los gustos de los usuarios. Esto afecta a la eficacia de los sistemas recomendadores, que no van a poder ofrecer productos nuevos que puedan interesarle al usuario.

También tenemos el problema de la aparición de nuevos usuarios. Como en el caso de los sistemas de recomendación basados en filtrado colaborativo, es necesario que un usuario interactúe con los productos de la plataforma para poder crear su perfil de usuario.

Por último, tenemos las restricciones sobre el contenido. Obtener el contenido de un producto puede ser costoso si el dominio es complejo y grande. Sería necesario automatizar este proceso de alguna forma para hacer más eficiente la recomendación.

2.2.3. Sistemas de recomendación basados en conocimiento

Tanto los sistemas de recomendación basados en contenido como los basados en filtrado colaborativo necesitan gran cantidad de información sobre las interacciones llevadas a cabo por los usuarios del sistema. Como se ha comentado, esto supone un problema cuando aparecen nuevos productos en la plataforma o nuevos usuarios. Los sistemas de recomendación basados en contenido mejoran la eficacia cuando aparecen nuevos productos, pero aún así, siguen produciéndose recomendaciones poco eficaces en este tipo de escenarios. Además, estos métodos son poco adecuados para productos muy personalizados o productos con los que los usuarios no suelen interactuar, como pueden ser, por ejemplo, objetos de lujo o actividades de ocio o turísticas. También, cuando los productos son complejos de describir, es más difícil poder encontrar relaciones entre los productos o con el perfil del usuario [2].

Los sistemas de recomendación basados en conocimiento sirven para resolver las restricciones de los anteriores tipos de recomendadores, es decir, solucionan el problema de las

Tipo de recomendador	Objetivos del usuario	Input
Filtrado colaborativo	Proporciona recomendaciones calculadas a partir de las valoraciones e interacciones de todos los usuarios (incluido el usuario objetivo) con los productos del sistema.	Valoraciones del usuario + valoraciones del resto de usuarios
Basado en contenido	Proporciona recomendaciones basándose en las descripciones o atributos de los productos del sistema, favoreciendo los gustos del usuario objetivo.	Valoraciones del usuario + atributos de los productos
Basado en conocimiento	Proporciona recomendaciones en base a las restricciones del usuario sobre el conjunto de los productos del sistema.	Restricciones del usuario + atributos de los productos + conocimiento del dominio

Tabla 2.1: Diferencias entre los principales tipos de sistemas de recomendación [2]

pocas interacciones o valoraciones con ciertos productos. Estos sistemas de recomendación tienen en cuenta los requisitos de los usuarios (que realizan de forma explícita) para hacer las recomendaciones. Es decir, utilizan todo el conocimiento de los usuarios, productos y dominio para hacer las recomendaciones, haciendo más personales los resultados obtenidos [41]. Uno de los aspectos más característicos de estos recomendadores es que el usuario tiene un gran control sobre las recomendaciones que se le van a hacer por la necesidad de establecer sus propios filtros o requisitos sobre el amplio dominio que se le plantea. Aún así, estos sistemas también pueden tener sus desventajas, cuando los dominios del sistema son complejos o los requisitos que ponen los usuarios no se entienden. Se pueden ver las diferencias entre los tres recomendadores estudiados en la Tabla 2.1.

En general, los sistemas de recomendación basados en conocimiento son muy apropiados en las siguientes situaciones [2]:

- Cuando los clientes quieren especificar los requisitos en las recomendaciones.
- Cuando es difícil obtener las valoraciones o información de interacciones de un deter-

minado tipo de producto debido a la amplia complejidad de dominio del sistema con respecto a la variedad de tipos y opciones de productos.

- Cuando se hacen recomendaciones sobre productos que evolucionan muy rápidamente. Por ejemplo, en productos como ordenadores o coches, es difícil que las valoraciones que han sido realizadas por un usuario en un determinado momento tengan validez pasado un cierto tiempo. Los productos evolucionan tan rápidamente que los usuarios no quieren adquirir productos con características anticuadas, aunque en el pasado les hayan gustado.

Aunque estas características son comunes para todos los sistemas de recomendación basados en conocimiento, existe una clasificación realizada por Jannach et al. [20], por la cual se especifican dos tipos de este modelo de sistemas recomendadores [41]. Ambos tipos utilizan un proceso de recomendación similar por el cual el usuario determina ciertos requisitos de los productos que quiere adquirir, y a partir de ello se generan las recomendaciones. La diferencia entre ambos tipos de recomendadores se encuentra en la forma de utilizar el conocimiento.

Por un lado, existen los recomendadores basados en casos [48]. Éstos se basan en la utilización de métricas para recomendar productos similares a los ya valorados por el usuario. El proceso de recomendación de estos sistemas se ilustra en la Figura 2.1.

Por otro lado, se encuentran los basados en restricciones [16, 56]. Se basan en un conjunto de reglas definidas para hacer las recomendaciones y en el uso de estas reglas para determinar los productos a recomendar (que serán aquellos que cumplan las reglas). El proceso de recomendación de estos sistemas se encuentra en la Figura 2.2.

También se puede hacer una clasificación de estos sistemas si tenemos en cuenta cómo

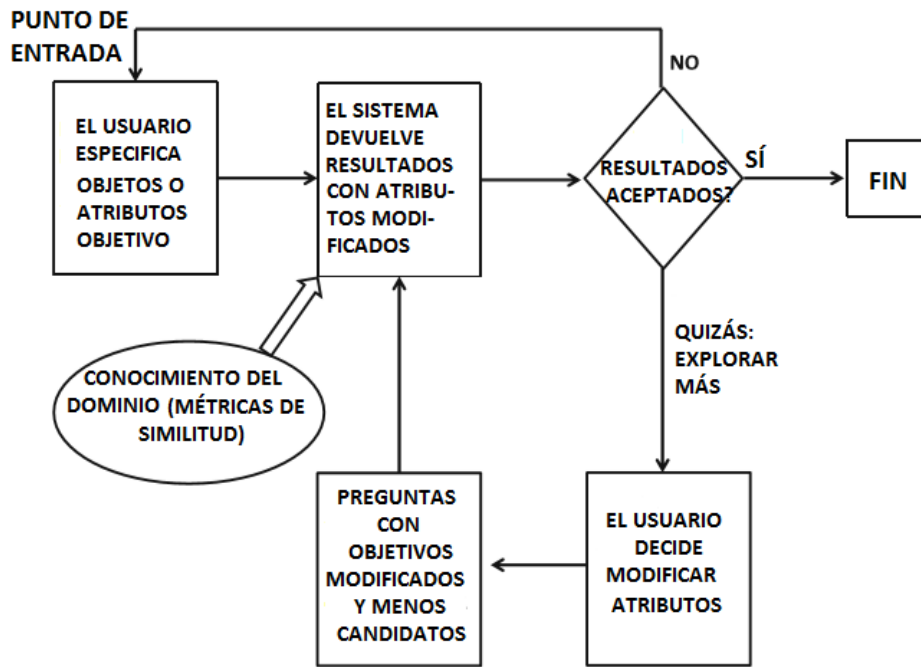


Figura 2.1: Proceso de recomendación de un recomendador basado en casos [2].

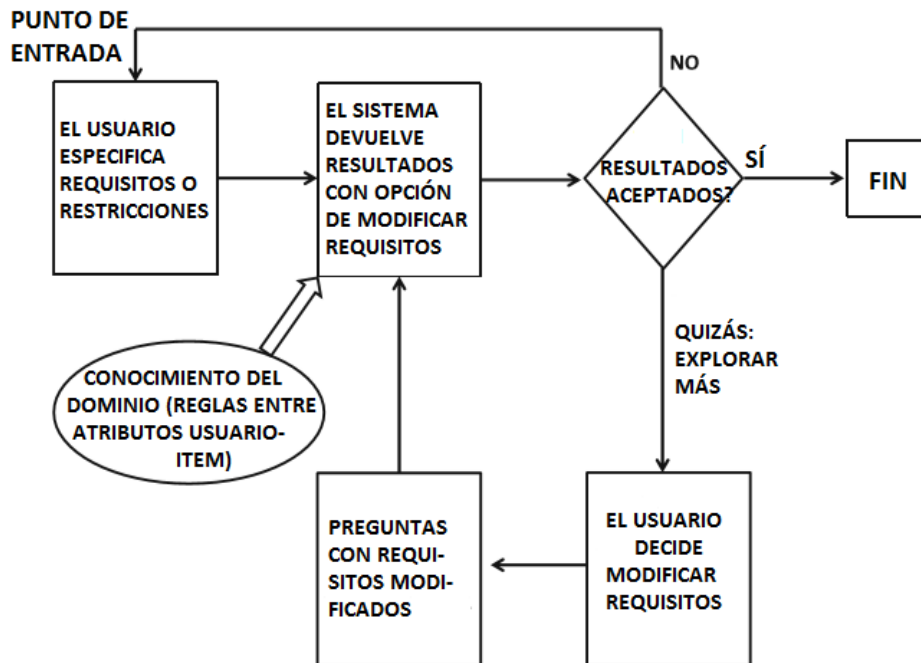


Figura 2.2: Proceso de recomendación de un recomendador basado en restricciones [2].

interactúan los usuarios con el sistema [2]. Existen los sistemas conversacionales, por los cuales el sistema va preguntando en bucle al usuario por las restricciones que quiere imponer en su búsqueda. Se usan en escenarios en los que el dominio del producto es muy amplio o complejo. También existen los sistemas basados en búsquedas, en los cuáles el sistema proporciona al usuario un conjunto de preguntas que debe responder para realizar las restricciones. Y por último, se encuentran los sistemas basados en navegación. Consiste en que el usuario establece las restricciones o filtros una vez se ha hecho la recomendación. De esta forma, se van estableciendo recomendaciones iterativas mientras el usuario va especificando sus requisitos.

En este caso, las debilidades de estos sistemas son: la dificultad en el conocimiento del problema, la debilidad más destacable de todas porque es imprescindible conocer el dominio del problema para hacer recomendaciones lo más eficaces posibles; la sobreespecialización de las recomendaciones, ya que es probable encontrarnos recomendaciones similares para un determinado usuario; y el problema del conjunto vacío, que consiste en que no se obtenga ningún resultado si el usuario incluye demasiadas restricciones en la búsqueda del producto [41].

Con el fin de superar las debilidades de los sistemas de recomendación detallados anteriormente, surgió un nuevo sistema de recomendación, el híbrido, que se comentará en la siguiente sección.

2.2.4. Sistemas de recomendación híbridos

En las anteriores secciones, se ha revisado el funcionamiento de los tres principales tipos de sistemas de recomendación. Cada tipo de sistema tiene sus propias debilidades y algunas son resueltas por los otros recomendadores. De la idea de combinar los diferentes sistemas de

recomendación para eliminar todas las debilidades vistas anteriormente surgen los sistemas de recomendación híbridos [2].

Los sistemas híbridos son los sistemas de recomendación que mezclan algunas (o todas) de las técnicas anteriores para obtener sus beneficios y vencer sus debilidades. La información para generar las recomendaciones puede obtenerse de las diferentes fuentes de datos y se puede usar la eficiencia de los diferentes algoritmos de recomendación para hacer el sistema más robusto. La decisión más importante es decidir qué sistemas mezclar y qué información va a utilizar cada uno [41]. En muchas ocasiones, los sistemas de recomendación híbridos combinan diferentes tipos de recomendadores, pero también se pueden combinar modelos del mismo tipo.

Existen tres formas de diseñar sistemas de recomendación híbridos [2]:

1. **Diseño en conjunto.** Se aplican diferentes algoritmos de recomendación sobre un único *input* para generar un único *output* más robusto. Por ejemplo, se pueden combinar los *outputs* obtenidos de un recomendador de filtrado colaborativo y de un recomendador basado en contenido para obtener una única lista de recomendación. Existen dos tipos de diseños dentro de este tipo. Uno es el diseño en paralelo, que combina todos los resultados obtenidos de aplicar cada sistema de recomendación sobre el mismo *input* para obtener un único *output* (Figura 2.3). El otro es el diseño secuencial que lo que hace es aplicar un sistema recomendador sobre un *input* y utilizar ese *output* como *input* del siguiente sistema recomendador, y así sucesivamente, hasta aplicar todos los sistemas recomendadores y por último combinar todos los *outputs* obtenidos en un único *output* (Figura 2.4).
2. **Diseño monolítico.** Se utiliza un único algoritmo de recomendación sobre varios *inputs* para generar un único *output*.

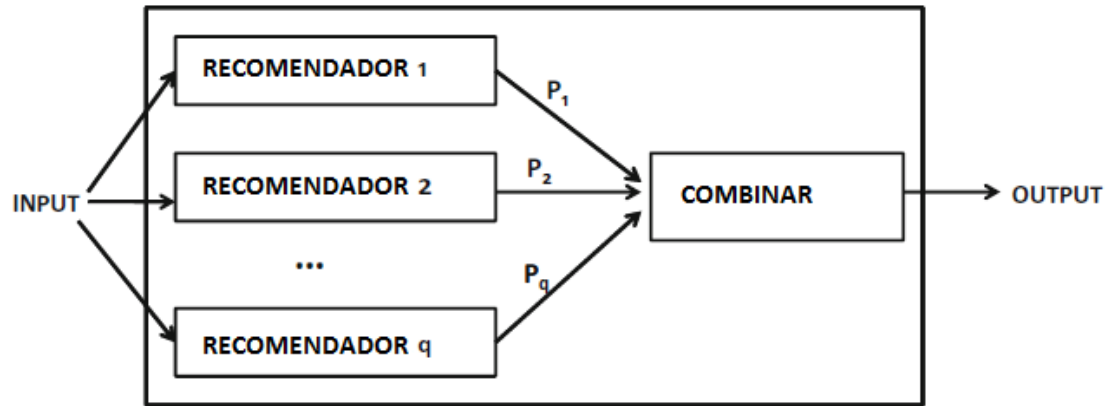


Figura 2.3: Diseño paralelo de un recomendador híbrido [2].

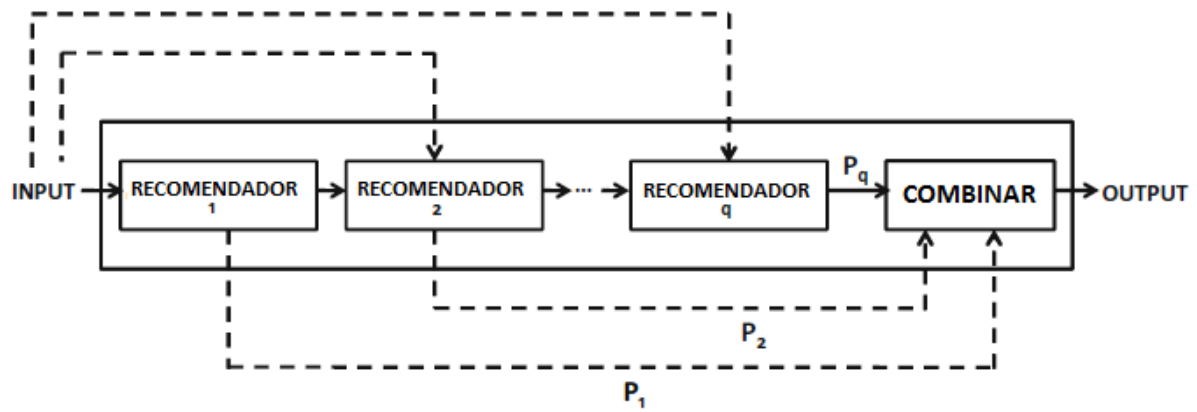


Figura 2.4: Diseño secuencial de un recomendador híbrido [2].

3. **Sistemas mezclados.** Se utilizan diferentes algoritmos de recomendación sobre un único *input*, como en el caso del diseño en conjunto, pero se obtienen varios *outputs*.

Dentro de estos tres tipos de diseños, existen siete técnicas diferentes para combinar estrategias de recomendación [9]. (En la Figura 2.5 se muestra la clasificación de estas técnicas en los tres tipos de diseños vistos antes):

- **Ponderación:** las valoraciones otorgadas por los diferentes tipos de recomendadores se agregan a través de algún tipo de esquema. El producto con mayor valoración será el que se recomienda al usuario. Cada recomendador puede tener diferente peso en la valoración.
- **Cascada:** se utiliza un recomendador sobre el conjunto inicial, obteniendo un nuevo conjunto. A partir de éste, se aplica el otro recomendador y así sucesivamente hasta llegar al resultado final.
- **Conmutación:** cuando se cumplen ciertas condiciones, se aplica un recomendador, y si no, el resto de métodos o recomendadores.
- **Mixto:** el resultado obtenido es un conjunto de los resultados obtenidos con todas las técnicas de recomendación usadas.
- **Combinación de características:** los datos utilizados son la unión de todos los datos que utilizan todos los sistemas. Después, se aplica un único algoritmo de recomendación sobre ese conjunto.
- **Incorporación de características:** uno de los métodos genera una recomendación, más una característica nueva (por ej. productos relacionados) para el producto, que va a ser integrada como una nueva característica para el resto de estrategias de recomendación.

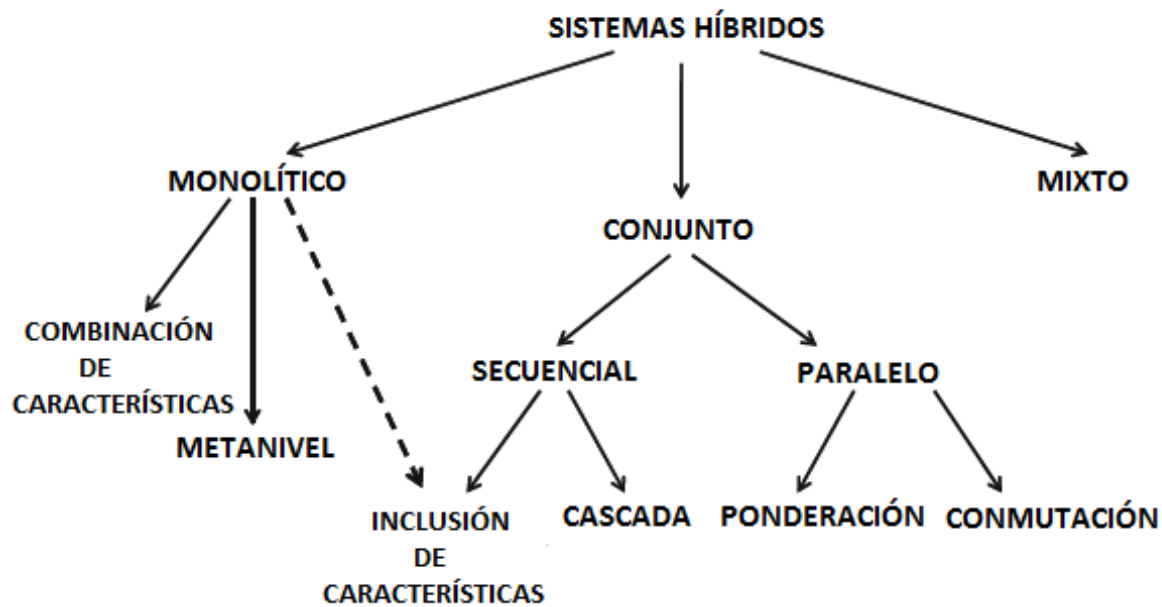


Figura 2.5: Clasificación de las técnicas de para combinar estrategias de recomendación en los tres tipos de diseños híbridos [2].

- **Metanivel:** el modelo completo generado por un recomendador es usado para el resto de sistemas recomendadores. Se suele utilizar típicamente la combinación de filtrado colaborativo y recomendadores basados en contenido.

Habiendo realizado ya una revisión de los tipos de sistemas de recomendación, se pueden tomar unas conclusiones con respecto a sus debilidades y los retos que quedan por delante en líneas de investigación en el campo. En la siguiente sección se va a proceder a detallar estas conclusiones.

2.3. Retos de los sistemas recomendadores y líneas de investigación

Gran parte del trabajo desarrollado en la investigación en sistemas de recomendación se ha centrado en obtener grandes resultados en las recomendaciones realizadas a los usuarios del sistema. Es decir, se ha hecho hincapié en proporcionar listas de productos que sean

muy útiles al usuario, de forma que la precisión de estas listas sea la mejor posible. Aún así, no es sólo importante la calidad de las recomendaciones proporcionadas, sino que también es importante cuál es la posterior relación que tienen los usuarios con esas recomendaciones [33]. Se tienen que tener en cuenta otros aspectos como la diversidad de los resultados, las recomendaciones a realizar para usuarios nuevos o poco habituales o la confianza de los usuarios en el sistema recomendador.

Teniendo en cuenta las debilidades que tienen actualmente los sistemas recomendadores, se pueden determinar cuáles son los principales problemas a resolver en el campo de investigación que nos ocupa [39]. Por ello, a continuación se van a recoger las debilidades (algunas nuevas y otras mencionadas anteriormente) como forma de ilustrar los retos a superar y líneas de investigación a desarrollar en el futuro de este campo. Son las siguientes:

- ***Cold start problem* o arranque en frío.** Así es como se llama al problema que surge cuando aparecen en el dominio del problema nuevos usuarios o productos de los cuales no existe información sobre ellos. Es necesario resolver este problema para hacer mejores recomendaciones a los nuevos usuarios debido a los problemas reiterados durante todo el capítulo.
- **Dispersión.** Para los usuarios poco habituales también es necesario mejorar las recomendaciones que se les proporcionan, ya que al tener poca información sobre ellos, es más difícil acertar en las recomendaciones.
- **Ataques al recomendador.** Se producen cuando una entidad, que conoce el sistema de recomendación, valora ciertos productos para manipular los resultados del sistema de recomendación.
- **Desconfianza de los usuarios en el sistema recomendador.** Al producirse estos ataques, los sistemas de recomendación no funcionan correctamente y es por eso por lo que aumenta la desconfianza de los usuarios con respecto a las recomendaciones

que el sistema les proporciona. En muchas ocasiones, el usuario ajusta sus propias valoraciones para que el recomendador sea más efectivo [2].

- **Privacidad.** Cuanta más información proporcionen los usuarios, el recomendador podrá obtener mejores resultados. Sin embargo, que los usuarios den demasiada información al sistema recomendador también puede provocar problemas de privacidad para esos usuarios, quienes pueden obtener información sensible de los mismos o incluso utilizar estos datos en su propio beneficio (por ejemplo, vendiéndolos a terceros).
- **Ausencia de explicación en el proceso de recomendación.** Los sistemas de recomendación, en general, no explican por qué los productos recomendados han sido recomendados y porque el resto de productos no.
- **Recomendaciones uniformes (Efecto portafolio):** todas las recomendaciones son similares entre sí, por lo que no resultan de utilidad al usuario. Es importante proporcionar una lista de recomendación de productos con cierta variedad, ya que los usuarios pueden cambiar de gustos a lo largo del tiempo o tener gustos diferentes sobre un determinado tema. La diversidad en las recomendaciones tiene que tener como objetivo generar recomendaciones útiles al usuario, pero que tenga una cierta variedad para cubrir los intereses del usuario y que tengan diferentes posibilidades de elección.
- **Falta de contextualización de las recomendaciones.** El contexto puede influir de manera significativa sobre qué necesitan los usuarios de las recomendaciones proporcionadas. El contexto es el conjunto de factores que tienen impacto sobre cómo se van a seleccionar y clasificar los productos en las recomendaciones [20]. Algunos de estos factores importantes pueden ser el lugar en el que recibe las recomendaciones el usuario, con quién recibe esas recomendaciones o qué recursos están cerca del usuario en ese momento [46].
- **Mejora de la interacción del usuario con el recomendador.** En general, los

sistemas de recomendación funcionan tras la interacción de un usuario con uno o varios productos, o tras una consulta del usuario al sistema para pedir recomendaciones. Existen usuarios que no tienen por qué conocer el dominio del problema, por lo que les puede resultar difícil hacer una consulta al sistema. De esto proviene la necesidad de investigar en el campo de desarrollo de nuevas interacciones entre el usuario y el sistema, de forma que el comportamiento reactivo sea sustituido por un comportamiento proactivo por parte del recomendador, y sea el propio sistema el que recomiende productos al usuario sin que tenga el propio usuario tenga que hacer una petición o una interacción. Existen ya recomendadores que proporcionan propuestas intermedias entre los sistemas reactivos y los proactivos: los sistemas de conversación (las estrategias ya mencionadas: la de navegación-por-propuesta o la de navegación-por-pregunta) [41].

- **Aprender a valorar.** De las recomendaciones realizadas, cada usuario tiene en cuenta sólo las k mejores. Se necesita, por tanto, conseguir que las recomendaciones más adecuadas para cada usuario estén siempre en los primeros puestos de la lista de productos recomendados [2].
- **Recomendaciones para grupos.** Hace falta invertir más investigación en recomendaciones grupales, aquellas en las que el sistema tiene que generar recomendaciones para un conjunto de personas y satisfacer las necesidades y gustos de cada una de ellas, minimizando el error para el menor número de personas [2].
- **Sistemas de recomendación con múltiples criterios.** También es interesante el hecho de tener sistemas de recomendación que valoren diferentes factores de un determinado producto y en función de eso, recomendar productos según los diferentes factores que tengan los productos con los factores más interesantes para un producto [2].

A pesar de estas debilidades y de la cantidad de trabajo que aún queda por realizar, los

sistemas de recomendación son unos de los sistemas más necesarios actualmente en muchas plataformas de trabajo, de entretenimiento, consumo y comercio, aprendizaje y enseñanza. En la siguiente sección, se van a comentar algunos de los ejemplos de sistemas de recomendación más utilizados e importantes actualmente.

2.4. Aplicación de los sistemas de recomendación

Existe una gran diversidad de tipos de productos a recomendar. El uso mayoritario que se hace de los sistemas de recomendación es en el campo del comercio electrónico, en la compra-venta de productos, libros, películas, viajes y otro tipo de servicios. Aunque esta es la utilidad típica de los recomendadores, lo cierto es que su uso se ha ampliado a la recomendación de otro tipo de productos y en otros contextos. Por ejemplo en la recomendación de personas (en una red social como Facebook⁴ o LinkedIn⁵), a la cual se le llama recomendación recíproca, o en el área de la publicidad computacional, que proporciona recomendaciones de productos a usuarios al realizar búsquedas, como en las búsquedas de Google⁶. También existen otras plataformas como Youtube⁷, donde se recomiendan vídeos, IMDb⁸, donde los ítems son películas o Tripadvisor⁹, que recomienda hoteles, viajes, etc.

Con el objetivo de ilustrar las utilidades que los sistemas de recomendación pueden tener en múltiples plataformas, se van a mostrar ejemplos de algunas aplicaciones muy conocidas hoy en día que utilizan sistemas de recomendación para mejorar la experiencia de usuario de estas aplicaciones.

⁴<https://www.facebook.com/>

⁵<https://www.linkedin.com/>

⁶<https://www.google.es/>

⁷<https://www.youtube.com/>

⁸<http://www.imdb.com/>

⁹<https://www.tripadvisor.es/>

Por un lado, tenemos el sistema recomendador de GroupLens¹⁰, el cual fue uno de los pioneros en investigación en sistemas de recomendación en filtrado colaborativo. Es utilizado en los recomendadores *BookLens* y *MovieLens*, para recomendación de libros y películas respectivamente, esta última con un conjunto de datos de dimensiones enormes.

El sistema de recomendación de Amazon¹¹ fue uno de los primeros sistemas de recomendación en ser usado de manera comercial [27]. Es un sistema recomendador que tiene en cuenta las valoraciones de los usuarios, el comportamiento de los usuarios al comprar y también su comportamiento al navegar por la web (*feedback* explícito e implícito, respectivamente). Muchos sistemas de recomendación de tiendas online procesan ambos tipos de recomendación para generar la lista de productos recomendados, ya que se obtiene información relevante de las dos fuentes.

El sistema de recomendación de Netflix¹², por su parte, permite a los usuarios obtener nuevas propuestas de series y películas para consumir en *streaming*. El sistema permite a los usuarios hacer valoraciones con una escala de uno a cinco y además proporciona explicaciones efectivas de por qué esos productos han sido recomendados. El sistema de recomendación de Netflix ha contribuido en el desarrollo de investigación en el campo gracias al *Netflix Prize contest* [5], concurso celebrado por el cual los concursantes aportaban su propio algoritmo de filtrado colaborativo sobre un conjunto de datos de la base de datos de Netflix.

En el caso del sistema de personalización de las noticias de Google, el tipo de producto que se recomienda son noticias. Las recomendaciones se basan en el historial de *clicks* del usuario al que se recomiendan las noticias. La información por tanto se trata como *feedback* implícito, porque no son los propios usuarios los que realizan las valoraciones de las noticias.

¹⁰<https://grouplens.org/>

¹¹<https://www.amazon.es/>

¹²<https://www.netflix.com/es/>

Por último, destaca el sistema de recomendación de la red social Facebook¹³. Se utiliza como método para crear nuevas conexiones entre los usuarios de la red social, y así aumentar la conectividad entre los usuarios y la cantidad de información de cada uno de ellos.

2.5. Conclusiones

Los sistemas de recomendación, por tanto, son unos de los sistemas más necesarios en la Web de hoy en día, en concreto en aquellas páginas o plataformas que están orientadas al consumo de cualquier tipo de producto. Se pueden aplicar a plataformas de diferentes ámbitos, y en todas ellas este tipo de sistemas son necesarios para poder mejorar la experiencia de los usuarios y proporcionarles ayuda a la hora de consumir nuevos productos. Al diseñar un sistema recomendador, es importante conocer qué tipos de recomendadores existen para poder adecuar las necesidades de los usuarios a nuestro sistema. Los sistemas de recomendación basados en filtrado colaborativo son adecuados para aquellas plataformas donde los usuarios valoran productos, mientras que los basados en conocimiento o en contenido van a funcionar muy eficazmente en plataformas donde los productos tengan descripciones muy completas. Sin embargo, no todas las plataformas en las que sería adecuado implantar un sistema de recomendación cumplen estas características. Por ejemplo, los jueces en línea, donde los usuarios no valoran los problemas que realizan y donde las descripciones de los ejercicios suelen ser muy pobres (apenas unas etiquetas que definan el problema). Debido a este problema, estos tipos de recomendadores no son los más adecuados para aplicar sobre jueces en línea.

En este Trabajo Fin de Máster se proponen, como alternativa para los jueces en línea, métodos de recomendación basados en grafos de interacción, que van a proporcionar las

¹³<https://www.facebook.com/>

recomendaciones a los usuarios gracias a la utilización de grafos como modelos de representación de las interacciones entre los usuarios con los problemas contenidos en el juez. El diseño de los recomendadores propuestos se va a detallar en el siguiente capítulo.

Capítulo 3

Métodos de recomendación para jueces en línea

Los jueces en línea son plataformas que utilizan los estudiantes de informática y ciencias de la computación para estudiar y mejorar su capacidad de programar. Contienen ejercicios de estructuras de datos, algoritmos y programación que los estudiantes pueden realizar y entregar al mismo juez para que evalúe la solución que han realizado para un determinado problema y les proporcione un veredicto. Por tanto, existen interacciones entre los usuarios y la propia plataforma. Estas interacciones se pueden representar como un grafo.

Los grafos pueden representar multitud de sistemas sociales, biológicos o sistemas de información de los que se pueden obtener beneficios conociendo la información que contienen [29]. El proceso de recomendación basado en grafos está fundamentado en la idea de representar la información de un sistema como si fuese un grafo, de forma que los nodos del grafo sean los usuarios y/o productos del sistema, y los enlaces representen que hay una relación entre los dos nodos que se unen a través de ese enlace. A partir de este grafo se pueden predecir nuevos enlaces o enlaces perdidos, lo cual puede ser utilizado como forma de hacer recomendaciones a los usuarios del sistema. Las técnicas que permiten hacer estas predicciones se llaman técnicas de predicción de enlaces o *link prediction*.

En este capítulo se van a proponer un conjunto de métodos de recomendación para los jueces en línea. Para abordarlo, primero se va a hacer una breve introducción sobre qué son los jueces en línea y la necesidad de incluir sistemas de recomendación en este tipo de sistemas (sección 3.1). Después, se va a hacer un estudio detallado sobre las técnicas de predicción de enlaces (sección 3.2). Por último, se va a proceder a explicar cómo se han diseñado los algoritmos de recomendación para jueces en línea basados en grafos y en técnicas de predicción de enlaces (sección 3.3).

3.1. Jueces en línea

En la actualidad, el desarrollo de las nuevas tecnologías es constante, lo que ha permitido la inclusión de estas tecnologías en los procesos de aprendizaje, también durante la formación universitaria. La inclusión de TICs (Tecnologías de la Información y la Comunicación) aporta innumerables ventajas con respecto a la educación universitaria tradicional. Entre estas ventajas, se puede destacar la flexibilidad, de modo que los alumnos puedan acceder a los contenidos educativos en cualquier momento y lugar, disponiendo simplemente de un ordenador o un dispositivo móvil para poder acceder a estos contenidos. Es decir, el acceso a la información es mucho más rápido y productivo y se puede consultar la información cuando se necesite. Las relaciones entre docente y estudiante también cambian. El intercambio de información y la colaboración entre ellos son más fluidos, mejorando el aprendizaje de los estudiantes de forma que sea mucho más personalizado. Como consecuencia, mejora el interés de los estudiantes y, por tanto, también su rendimiento, tanto por esta mejora en el acceso a los contenidos y la comunicación con el profesor, como por los nuevos métodos educativos aplicados, más cercanos a las nuevas generaciones y su contexto social actual [49].

Existen numerosas tecnologías que han surgido como métodos de enseñanza en las universidades. Una de estas tecnologías son los jueces en línea. Los jueces en línea (*online judges*)

son plataformas web que almacenan gran cantidad de ejercicios de programación, con el objetivo de que los estudiantes practiquen con estos ejercicios y mejoren sus habilidades en programación. Aunque los jueces en línea son utilizados por estudiantes universitarios de informática y ciencias de la computación, entre los cuáles se están haciendo cada vez más populares, estas plataformas surgieron como medios para celebrar concursos de programación. Aportan imparcialidad y reduce el trabajo de las personas que tendrían que juzgar las respuestas de esos participantes [24]. Incluso últimamente se utilizan jueces en línea como HackerEarth¹ para selección de candidatos en empresas como Amazon o Google.

Los jueces en línea funcionan como repositorios de ejercicios que los alumnos pueden realizar y los cuáles van a ser corregidos por un juez automático. Los estudiantes leen los enunciados de los ejercicios publicados en la web del juez en línea, programan la solución al ejercicio y después suben la solución (el código programado) al sitio web del juez. Lo que hace el sistema es compilar esa solución entregada por el estudiante y ejecutarla sobre los diferentes casos de prueba que tiene el juez. Las salidas textuales generadas son comparadas con la salida textual de la solución correcta que tiene el juez y, en función de esto, se genera un veredicto. Los posibles veredictos que puede otorgar un juez en línea a un estudiante que le haya entregado una solución varían según el juez en línea, aunque suelen ser muy parecidos, si no los mismos. En general, los veredictos principales que podemos encontrar en cualquier juez en línea son:

- *Accepted* (AC), cuando la solución es correcta.
- *Presentation Error* (PE), cuando la solución es correcta pero el formato de la salida de texto no es exactamente igual a la solución del juez.
- *Wrong Answer* (WA), cuando el programa no funciona correctamente.
- *Compile Error* (CE), si la solución subida no compila.

¹<https://www.hackerearth.com/>

- *Runtime Error* (RE), si la solución falla durante la ejecución de la misma.
- *Time Limit Exceeded* (TL), cuando el programa tarda en ejecutarse demasiado tiempo.
- *Memory Limit Exceeded* (ML), cuando el programa consume más memoria de lo que permite el juez.
- *Output Limit Exceeded* (OL), si la salida de texto del programa es demasiado larga.

El problema de los jueces en línea es que no suelen tener ningún tipo de mecanismo que permita a los usuarios reconocer cuáles son los problemas que más les interesa realizar. Los alumnos sólo pueden orientarse por los enunciados de los ejercicios y por las categorías en las que suelen estar organizados los problemas dentro del sistema o los *tags* asociados a cada problema que, en general, son las estructuras de datos a utilizar o los conceptos a resolver en ese ejercicio. En ocasiones, el único mecanismo que permite determinar al usuario qué problemas realizar es el uso del *Global Ranking Method* por parte del juez en línea, que lo que hace es recomendar a los usuarios nuevos en el sistema el problema que más soluciones correctas ha recibido por parte de los demás usuarios del sistema y que no haya sido realizado por ese usuario en concreto. Es un método que carece de personalización: se recomienda a casi todos los usuarios el mismo problema, el cual no puede ajustarse a los intereses y necesidades de un usuario concreto. Como consecuencia, los jueces en línea necesitan incluir recomendadores para mejorar la experiencia de sus usuarios y personalizar la recomendación de ejercicios, ayudándoles en su objetivo de progresar en materia de programación.

Existen algunos jueces en línea disponibles actualmente [30]. El UVa Online Judge², el juez de la Universidad de Valladolid, es uno de los más conocidos, utilizados y mejor implantados en la actualidad. También, destaca *Jutge.org*³, desarrollado por la Universidad

²<https://uva.onlinejudge.org>

³<https://jutge.org/>

Politécnica de Cataluña (UPC). Entre otros usos, *Jutge.org* se utiliza en la Olimpiada Informática Española⁴, concurso de informática a nivel nacional para alumnos de secundaria y formación profesional que actualmente está gestionado por la Universidad Politécnica de Cataluña (UPC) y la Facultad de Informática de Barcelona (FIB). Mencionar también que existen otros muchos jueces como *Codeforces*⁵, el *Light OJ* o *Light Online Judge*⁶ y el *Sphere online judge*⁷. Por último, destacar la existencia del juez en línea Acepta el Reto⁸, el juez de la Universidad Complutense de Madrid, desarrollado por profesores de la Facultad de Informática de esa misma universidad. Actualmente es utilizado por los alumnos de esta facultad para avanzar en sus asignaturas y también como medio de celebración de concursos de programación. Este es el juez en línea sobre el que se va a desarrollar la evaluación de los recomendadores propuestos en este Trabajo Fin de Máster. Se hablará sobre él más en profundidad en el siguiente capítulo, que es el que tratará el proceso de experimentación y evaluación con Acepta el Reto.

En la siguiente sección se va a hacer un estudio detallado de las técnicas de predicción de enlaces sobre grafos, en las que nos vamos a basar para generar las recomendaciones de los sistemas propuestos.

3.2. Técnicas de predicción de enlaces

Analizar grafos para obtener la información que nos puede interesar de ese grafo es una tarea compleja debido a que los grafos que representan sistemas (sociales, biológicos, etc.) están incompletos (toda la información del sistema no está representada en el grafo) y porque los grafos son dinámicas (están en continuo cambio: aparecen o desaparecen nodos o enlaces

⁴<https://olimpiada-informatica.org/>

⁵<http://codeforces.com/>

⁶<http://www.lightoj.com>

⁷<http://www.spoj.com/>

⁸<https://www.aceptaelreto.com/>

constantemente) [54]. Existen estudios que analizan cómo se estructuran estos grafos [11], cómo se establecen las relaciones entre los elementos del grafo [7] o cómo evolucionan los grafos a lo largo del tiempo [14]. Predecir la evolución de un grafo supone uno de los mayores retos actualmente dentro de las ciencias de la computación por su complejidad y por la necesidad que ha surgido de analizar grafos de cualquier materia, como medicina, economía o gestión, con el objetivo de extraer información para, por ejemplo, predecir cómo va a desarrollarse una epidemia, capturar delincuentes o detectar fallos en redes de transporte [4]. En referencia a este problema, uno de los retos fundamentales que se necesita resolver es el problema de la predicción de enlaces o *link prediction*, necesario para conocer cómo va a evolucionar un grafo en base a su estado actual [29].

La predicción de enlaces es un conjunto de técnicas que engloban algoritmos que se aplican sobre grafos con el objetivo de predecir nuevos enlaces o enlaces que van a dejar de existir, así como descubrir contenido del sistema que aún no está contemplado en el grafo [26, 54]. Formalmente, se puede definir un grafo $G = \langle V, E \rangle$, donde V es el conjunto de nodos del grafo G , y E es el conjunto de enlaces de G dados dos instantes de tiempo t_0 y t_1 , siendo $t_0 < t_1$, los métodos de predicción de enlaces sirven para determinar $E_1 \setminus E_0$, donde E_0 es el conjunto de enlaces de G en el instante t_0 y E_1 , el conjunto de enlaces de G en el instante t_1 [40]. Se puede ver un ejemplo de un problema de predicción de enlaces en la Figura 3.1.

Existen principalmente dos estrategias para predecir la evolución de los enlaces de un grafo: las estrategias basadas en similitud y las basadas en aprendizaje [54]. Lo que hacen las estrategias basadas en similitud es medir la semejanza entre dos nodos x e y no conectados y, en función de esto, asignar una puntuación a ese par de nodos que va a indicar la probabilidad de que ambos nodos se conecten.

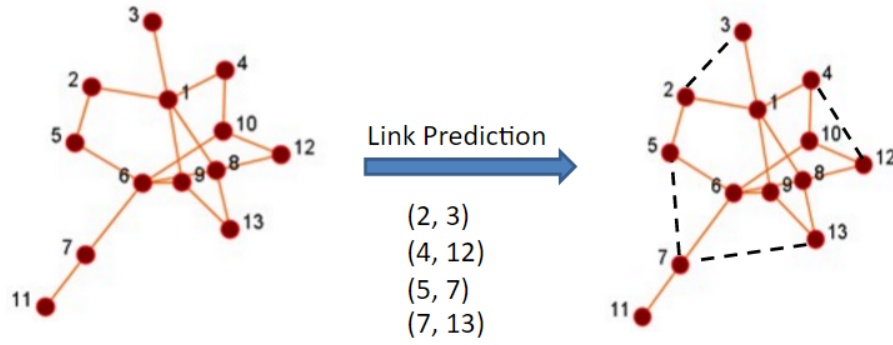


Figura 3.1: Ejemplo de un problema de predicción de enlaces [50]

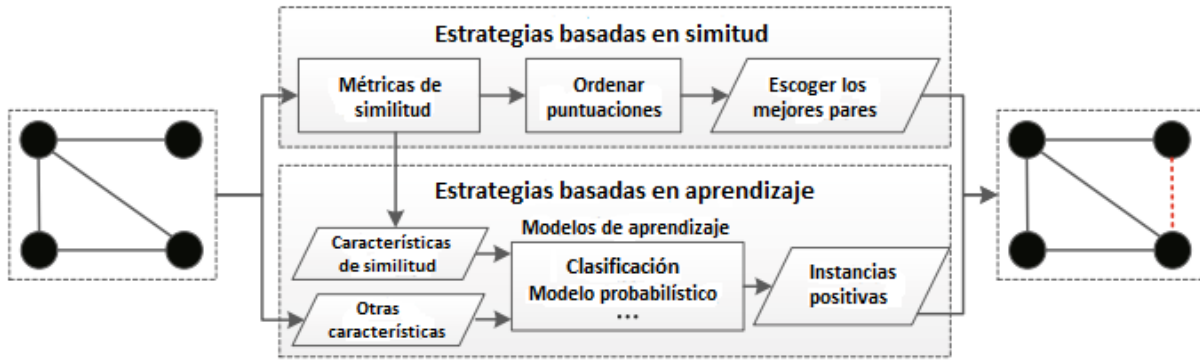


Figura 3.2: Diferencia de procesamiento entre los diferentes tipos de técnicas de predicción de enlaces [54]

Por el contrario, las estrategias basadas en aprendizaje tratan el problema de predicción de enlaces como tareas de clasificación binaria, por lo que se pueden utilizar modelos de aprendizaje automático, como el modelo de clasificación o el modelo probabilístico para resolver el problema [3]. Cada par de nodos del grafo es catalogado como positivo si existe probabilidad de que esos nodos se conecten o negativo en caso contrario. En la Figura 3.2 se observan las diferencias de procesamiento entre ambas técnicas.

A su vez, las estrategias o métricas basadas en similitud se clasifican en: métricas basadas en los nodos del grafo, métricas basadas en la topología del grafo y métricas basadas

en teorías sociales [54]. Las métricas basadas en los nodos del grafo presuponen que un par de nodos x y y no conectados dentro de un grafo tendrán más probabilidades de enlazarse cuanto más se parezcan entre ellos. Esta teoría está basada en la homofilia [54], tendencia social que tenemos las personas de relacionarnos con quien compartimos intereses, aficiones, ciudad de residencia, trabajo, etc.

Las métricas basadas en la topología del grafo, por el contrario, tienen en cuenta la estructura del grafo para medir la similitud entre los pares de nodos. Se dividen en tres tipos de métricas:

- **Métricas basadas en vecinos:** son las métricas en las que se tienen en cuenta los nodos vecinos de un par de nodos del grafo para obtener el valor de similitud de ese par de nodos. Se basa en el hecho social por el cual las personas tendemos a relacionarnos con las personas que más cerca tenemos.
- **Métricas basadas en caminos:** tienen en cuenta los caminos que existen entre los nodos del grafo para calcular el valor de similitud entre los nodos no conectados.
- **Métricas basadas en caminos aleatorios o *random walk*:** para conseguir un valor de similitud para un par de nodos no conectados del grafo se utilizan caminos aleatorios. Las interacciones que se crean en un sistema se representan con caminos aleatorios generados a partir de las probabilidades de conexión de los nodos a través de sus vecinos.

Por otro lado, las métricas basadas en teoría social son las que hacen las predicciones más efectivas, ya que no sólo tienen en cuenta la topología del grafo para generar las valoraciones de similitud, sino que también tienen en cuenta los aspectos sociales del sistema, lo que aporta información adicional a la que es extraída de la estructura del grafo. Un ejemplo es el modelo propuesto por Valverde-Rebaza et al. [51], por el cual se mezclan los conocimientos basados en la topología del grafo con información de la comunidad de usuarios, como sus

intereses y comportamientos para predecir futuras relaciones en Twitter.

Las estrategias basadas en aprendizaje también se pueden dividir en diversos tipos: clasificación basada en características, modelo de grafo probabilístico y factorización de matrices. La clasificación basada en características es el conjunto de métricas que determinan que se creará un enlace cuando el par de nodos no conectados pertenezca a la clase de nodos conectados, es decir, cumplirán las propiedades que también cumplen los nodos ya conectados del grafo.

Por su parte, el modelo probabilístico presupone que el grafo se adapta a un tipo de grafo concreto y a partir de ello, se define la probabilidad de que se formen enlaces entre los nodos no conectados. El modelo basado en factorización de matrices [34], trata el problema de predicción de enlaces como un problema de factorización de matrices, es decir, intenta encontrar características latentes en común entre los nodos para decidir si pueden enlazarse o no.

A pesar de la popularidad que han adquirido en los últimos años, tanto a nivel empresarial como a nivel universitario por la cantidad de aplicaciones y beneficios que tiene su uso, el problema de la predicción de enlaces es realmente complejo y aún quedan retos por superar en el futuro [26, 54]. Algunos de estos retos que van a dirigir la futura investigación en este campo son:

- **Predicción de enlaces temporal.** Este consiste en determinar los enlaces que se van a crear en el grafo, pero la dificultad está en predecirlo en un tiempo t [15, 35].
- **Predicción de enlaces en grafos heterogéneos.** Se trata de resolver el problema de predecir enlaces en grafos donde los nodos son de diferentes tipos y no del mismo (por ejemplo, usuarios y problemas) y donde se pueden establecer relaciones distintas (enlaces) entre los nodos del grafo, ya sean iguales o diferentes. Este tipo de redes son

las que representan de forma más fiel la realidad, por lo que es importante investigar y proponer soluciones sobre este problema. Este Trabajo Fin de Máster sigue esta línea de investigación.

- **Predicción de enlaces con enlaces activos e inactivos.** El problema está en identificar cuáles son los enlaces más activos, es decir, qué enlaces del grafo son los más recientemente usados, ya que son más importantes que los usados hace más tiempo.
- **Predicción de enlaces para enlaces que van a desaparecer.** La mayor parte de los estudios llevados a cabo en el campo de la predicción de enlaces ha sido para determinar nuevos enlaces que se iban a crear y, sin embargo, se ha obviado el hecho de que los enlaces de un grafo también pueden eliminarse, por ejemplo, aquellos enlaces que se han establecido entre nodos muy diferentes tienden a desaparecer habitualmente [38].
- **Predicción de enlaces para grafos dinámicos.** Los grafos son sistemas que están en continuo cambio haciendo una representación fiable de los sistemas sociales reales: el grafo evoluciona y los nodos y enlaces aparecen y desaparecen a lo largo del tiempo. Sin embargo, los métodos de predicción de enlaces aplicados sobre este tipo de representaciones aún no se han desarrollado en su totalidad, a excepción de unas pocas propuestas [25, 44].
- **Aplicación de teorías sociales sobre los grafos y los métodos de predicción de enlaces.** Como se ha explicado anteriormente, la incorporación de estas teorías va a dar mayor información en el problema de predicción de enlaces: son útiles para entender los mecanismos de formación de relaciones sociales. Por ello, deberían ser aplicadas más frecuentemente en los problemas de predicción de enlaces.
- **Obtención de evaluaciones justas y conjuntos de datos.** Encontrar conjuntos de datos adecuados con los que poder evaluar los resultados obtenidos con los métodos de predicción de enlaces es muy necesario para determinar si las métricas utilizadas

son las más acordes con los resultados que queremos obtener en nuestros estudios. Encontrar estos conjuntos de datos no resulta fácil cuando queremos obtener los resultados de aplicar las técnicas de predicción de enlaces en grafos muy grandes.

En la siguiente sección se va a proceder a detallar cómo se han generado los grafos de interacción en base a los cuáles hemos aplicado las técnicas de predicción de enlaces para generar los métodos de recomendación propuestos para jueces en línea.

3.3. Recomendación basada en grafos de interacción para jueces en línea

Los desarrolladores e investigadores en sistemas recomendadores tienen dos retos que vencer. Por un lado, tienen como reto representar la información sobre los usuarios y los productos, y por otro, construir un modelo lo suficientemente flexible como para incorporar diferentes métodos de recomendación. Una forma de lograr esto es representando la información del conjunto de datos del sistema como un grafo, y utilizar ese grafo para proporcionar las recomendaciones a los usuarios. Lo que se busca en el campo de la investigación en sistemas de recomendación es diseñar un modelo lo más genérico posible que pueda ser aplicado en el mayor número de plataformas [19].

Los modelos de sistema de recomendación basados en grafos utilizan los grafos como forma de representar las interacciones o relaciones de una plataforma. Los nodos de los grafos pueden ser los usuarios y/o los productos de esa plataforma, y los enlaces, las relaciones que se establecen entre los usuarios y/o los productos. Como se ha comentado en el capítulo anterior, son una buena alternativa para aquellas plataformas donde las valoraciones que hayan realizado los usuarios sean pocas o nulas o donde las descripciones de los productos sean inexistentes o muy breves.

A continuación se va a hacer la propuesta de diferentes métodos de recomendación para jueces en línea haciendo uso de grafos de interacción y de técnicas de predicción de enlaces. Estas propuestas están publicadas en los artículos presentados en el congreso *International Conference on Case-Based Reasoning* de 2016 [21] y de 2017 [10].

En un juez en línea, donde los elementos que van a interactuar entre sí van a ser usuarios y problemas, el grafo que representaría las interacciones que se establecen a través del juez sería un grafo bipartito [57], es decir, un grafo en el que un nodo puede pertenecer a dos clases diferentes: un nodo puede representar a un problema o puede representar a un usuario. Desde un punto de vista más formal representamos las interacciones existentes en un juez en línea a partir de dos clases de nodos: el conjunto de problemas disponibles en la plataforma $P = \{p_1, \dots, p_n\}$ y el conjunto de los usuarios de la plataforma $U = \{u_1, \dots, u_n\}$. Por lo tanto, los enlaces entre los nodos sólo se van a poder crear entre problemas y usuarios. Para crear un enlace entre un nodo usuario y un nodo problema se va a tener en cuenta si el usuario ha enviado una solución a un problema y, en caso de haberlo intentado, se puede tener en cuenta el veredicto que proporciona el juez a ese usuario sobre la solución que ha subido para ese problema. Suele ocurrir con frecuencia que, cuando los usuarios obtienen un veredicto negativo, vuelven a entregar una solución para ese problema del que han obtenido un mal resultado. También ocurre que algunos usuarios, aunque hayan recibido un resultado positivo para un determinado problema, vuelven a entregar versiones del código para ese mismo problema, con el objetivo de crear un código optimizado y subir su posición en el ranking de los alumnos que han realizado ese problema. Puede ocurrir incluso que en estas reentregas, los usuarios obtengan resultados negativos. El juez en línea no tiene esto en cuenta: va a guardar que el usuario ha obtenido un resultado positivo para ese problema en algún momento.

Por lo tanto, la relación que existe entre un problema y un usuario desde el punto de vista del juez en línea puede simplificarse de la siguiente forma:

- **Resuelto.** El usuario ha entregado varias soluciones para un determinado problema y al menos una de ellas es correcta. Se considera que una solución para un problema es correcta cuando el juez en línea proporciona como veredicto AC o PE. PE es una solución correcta a pesar de que la salida de texto generada por esa solución no sea la misma que posee el juez.
- **Intentado.** El usuario ha enviado una o más soluciones para un problema pero ninguna de ellas es correcta.
- **No intentado.** El usuario no ha enviado ninguna solución para ese problema.

Teniendo esto en cuenta, definimos que un nodo usuario u_i sólo se va a poder enlazar con un nodo problema p_j cuando el usuario u_i haya intentado resolver el problema p_j , o lo haya resuelto. A partir de esto, se puede definir una matriz de adyacencia $A = a_{ij}$ que especifique si un determinado usuario ha entregado una solución para un determinado problema, donde $a_{ij} = 1$ cuando el usuario u_i ha intentado resolver (o ha resuelto) el problema p_j .

El grafo bipartito creado se puede convertir en un grafo no bipartito, en el que todos los nodos van a pertenecer siempre al mismo conjunto, a través de un mecanismo llamado proyección de grafos bipartitos o *bipartite network projection* [58]. Esto se realiza con el objetivo de facilitar el problema de implementación de un método recomendador a partir de un grafo, ya que resulta más sencillo trabajar con un grafo donde los nodos pertenezcan a una única clase. Un ejemplo de cómo se lleva a cabo este mecanismo se puede observar en la Figura 3.3.

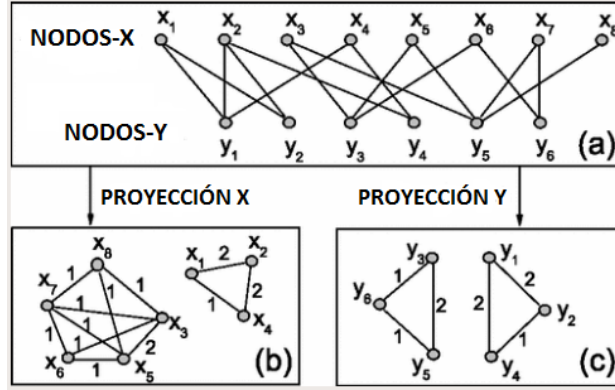


Figura 3.3: Transformación de un grafo bipartito en uno no bipartito a través del método *bipartite network projection* [58].

En este trabajo, se utiliza este método para convertir el grafo de nodos en dos posibles grafos no bipartitos, uno en el que los nodos pertenezcan sólo al conjunto de problemas y otro en el que los nodos sólo pertenezcan al conjunto de usuarios. Para evitar perder la información contenida en el grafo bipartito, definimos los pesos de los enlaces con información relevante que compartan ambos nodos. En las secciones 3.3.1 y 3.3.2 se explicarán, por separado, el método de recomendación aplicado a cada uno de los grafos.

Si el grafo es muy denso porque ha habido muchas interacciones en el juez en línea, es necesario aplicar algoritmos de filtrado de enlaces, de forma que sea más sencillo manejar el grafo con el que tenemos que trabajar.

Sobre el grafo ya construido y filtrado se pueden aplicar las técnicas de predicción de enlaces comentadas anteriormente con el objetivo de determinar las semejanzas existentes entre los nodos del grafo. En concreto, en este trabajo se han empleado métricas pertenecientes a los métodos basados en similitud. Con ellas se conseguirá una lista de los problemas más interesantes para los usuarios del juez. El objetivo de la utilización de varias métricas, en lugar de sólo una, es determinar qué métricas son las más prometedoras en los modelos de recomendación propuestos. En la sección 3.3.3 se detallarán las métricas utilizadas en

este trabajo.

Por lo tanto, a partir del grafo creado y de estas métricas, se puede crear una matriz de similitud Sim_m , que va a almacenar todos los valores de similitud de todas las posibles parejas de nodos del grafo, donde $Sim_m(i, j)$ va a contener el valor de similitud calculado con la métrica m para los nodos n_i y n_j . De esta forma, cuanto más parecido sea n_i a n_j según $Sim_m(i, j)$, más nos interesarán las interacciones de n_j en el juez para hacer las recomendaciones que involucren a n_i , y viceversa.

En este TFM se van a proponer y analizar los métodos de recomendación aplicados sobre los grafos de interacción que se obtienen al aplicar los métodos de proyección, uno con los problemas como nodos y otro con los usuarios como nodos. A continuación, se va a proceder a detallar ambos métodos, además de las distintas métricas de predicción de enlaces utilizadas y de los métodos de agregación propuestos para generar las recomendaciones.

3.3.1. Recomendación basada en el grafo de problemas

En la propuesta de recomendación basada en el grafo de problemas, vamos a partir de un grafo $G_p = \{N, E\}$, en el que $N = \{p \in P\}$ son problemas y $E = \{(p_i, p_j) | p_i, p_j \in P, i \neq j\}$ son los enlaces que conectan los nodos. Un enlace que une dos nodos p_i y p_j representa que existe al menos un usuario u_k que ha intentado o resuelto ambos problemas p_i y p_j . Los pesos de los enlaces van a ser el número de usuarios que han intentado o resuelto p_i y p_j . Si no existe ningún usuario que haya entregado una solución para ambos problemas, entonces el enlace no existirá.

Para hacer una recomendación basada en este grafo al usuario objetivo u_t , se van a implementar los siguientes pasos (Figura 3.4):

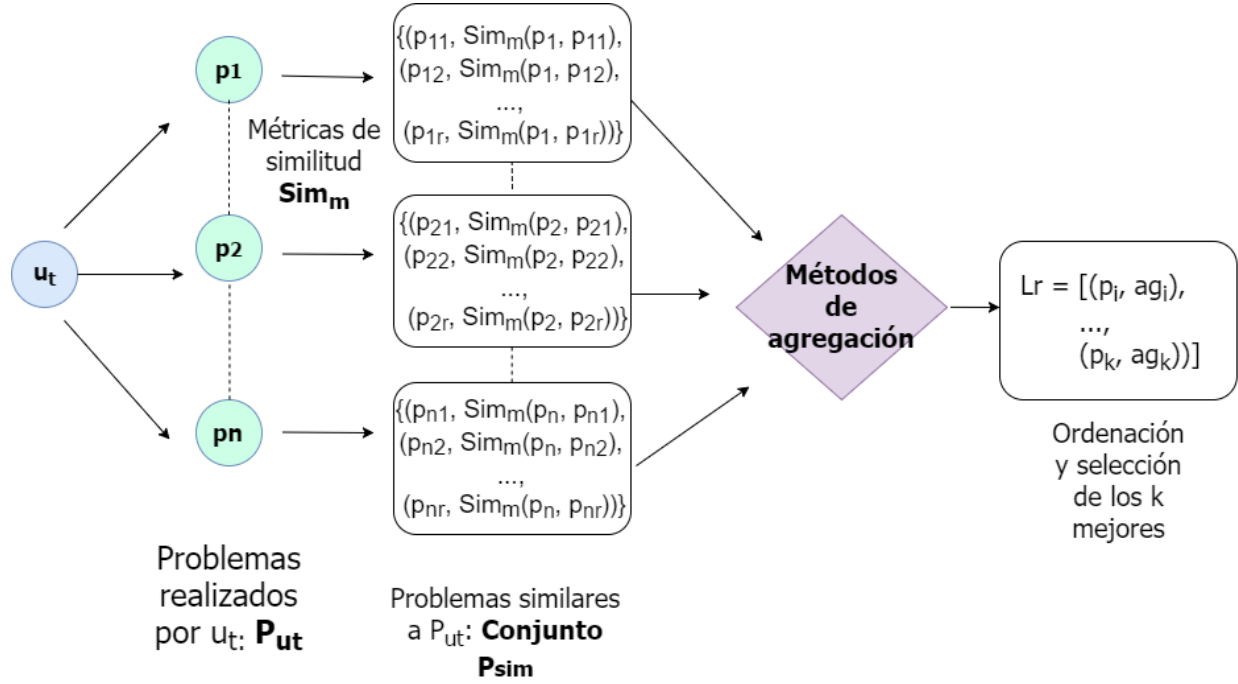


Figura 3.4: Generación de una recomendación basada en el grafo de problemas.

1. Calculamos el conjunto de problemas P_{u_t} , que es el conjunto que va a contener los problemas que ha realizado el usuario objetivo u_t .
2. Para cada $p_i \in P_{u_t}$ creamos un conjunto de problemas $P_{sim_{p_i}} = \{(p_j, s_j), \dots, (p_n, s_n)\}$ (siendo s_j el valor de similitud entre p_j y p_i : $s_j \equiv Sim_m(i, j)$), el cual va a ser el conjunto de problemas similares a p_i que no hayan sido realizados ya por el usuario objetivo u_t . Para simplificar, también se eliminarán problemas cuyo valor s_j sea menor a un valor umbral r_v que definamos.
3. Después, creamos una lista de problemas $L_r = [(p_j, ag_j), \dots, (p_x, ag_x)]$ que va a contener el valor de similitud agregado para aquellos problemas que aparecen en varios de los conjuntos calculados en el paso 2. ag_j es el valor de similitud agregado para el problema p_j obtenido a través de uno de los métodos de agregación propuestos en este TFM, que se detallarán más adelante en la sección 3.3.4.
4. Finalmente, ordenamos los problemas en L_r por su valor de similitud agregado y nos

quedamos con los k primeros problemas, que serán los que recomendemos al usuario objetivo u_t .

3.3.2. Recomendación basada en el grafo de usuarios

En esta ocasión, denotamos el grafo de usuarios del que vamos a partir para generar el recomendador como $G_u = \{N, E\}$, donde $N = \{u \in U\}$ son los usuarios de la plataforma, y $E = \{(u_i, u_j) | u_i, u_j \in U, i \neq j\}$ serán los enlaces formados entre los nodos. Un enlace entre dos usuarios u_i y u_j se crea cuando ambos usuarios han intentado o resuelto al menos un problema en común. Los pesos serán el número de problemas realizados en común por los usuarios que están conectados por ese enlace.

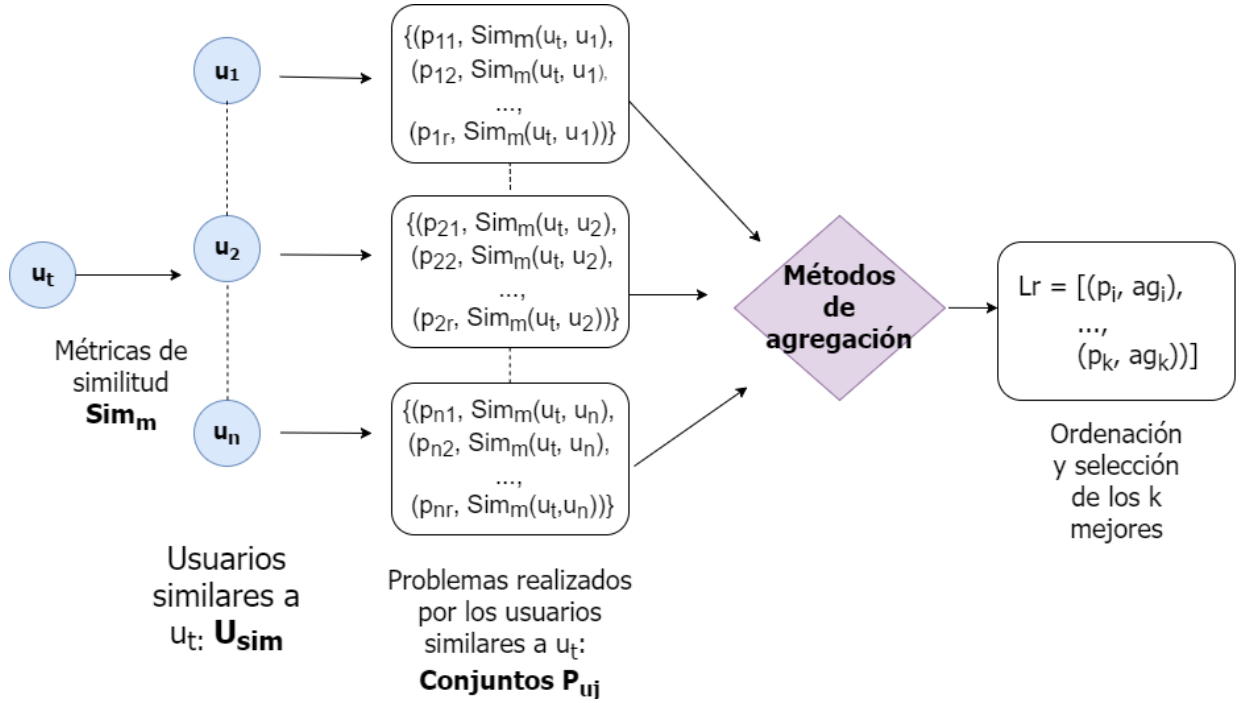


Figura 3.5: Generación de una recomendación basada en el grafo de usuarios.

Los pasos a seguir para realizar una recomendación basada en el grafo de usuarios a un usuario objetivo u_t son (Figura 3.5):

1. Generamos el conjunto de usuarios similares U_{sim} (fila t de la matriz Sim_m) al usuario objetivo u_t que tengan un valor de similitud superior a r_v .
2. Para cada $u_j \in U_{sim}$, creamos el conjunto de problemas $P_{u_j} = \{p_i, Sim_m(t, j)\}$ que ha intentado u_j y que no ha intentado el usuario objetivo u_t . En este caso, todos los problemas contenidos en un conjunto tienen el mismo valor de similitud.
3. Luego creamos la lista de problemas $L_r = [(p_1, ag_1) \dots (p_n, ag_n)]$, donde ag_j es el valor de similitud agregado para los problemas que aparecen repetidos en los conjuntos que se han creado en el paso anterior. Como en la recomendación basada en el grafo de problemas, para calcular ag_j se proponen diferentes métodos, que se detallarán en la siguiente sección 3.3.4.
4. Por último, se ordenan los problemas de la lista L_r por su valor de similitud agregado y seleccionamos los k primeros problemas para recomendar a u_t , ya que serán los mejores a recomendar para el usuario objetivo.

3.3.3. Métricas de similitud

Como hemos visto, para generar las recomendaciones necesitamos construir la matriz de similitud Sim_m la cual, como ya se ha comentado, se calcula a partir de la utilización de técnicas de predicción de enlaces. Esta matriz va a contener la información sobre la similitud que existe entre los diferentes pares de nodos del grafo. En esta sección se va a proceder a detallar cuáles han sido las métricas utilizadas y cómo calcula el valor de similitud cada una de ellas.

Para facilitar la comprensión de cómo funcionan estas métricas, hemos definido la siguiente notación:

- $N(x)$ es el conjunto de nodos vecinos del nodo x . Por lo tanto, $|N(x)|$ es el número de vecinos de x .
- $WD(x)$ es el *weighted node degree* del nodo x , es decir, la suma de los pesos de los enlaces que están conectados a x .
- A_{xy} es el peso del enlace que conecta los nodos x e y .

A continuación, se van a detallar las métricas utilizadas en nuestro trabajo teniendo en cuenta la notación definida. Algunas de estas métricas tienen dos versiones: la versión que usa los enlaces sin peso y la versión que usa los enlaces con peso.

- **Edge Weight o EW.** Es una métrica muy sencilla que determina el valor de similitud de dos nodos como el peso del enlace que une a ambos nodos. En caso de que dos nodos no estén conectados, el valor de similitud será 0.

$$EW(x, y) = A_{xy}$$

Existe una versión sin pesos de esta métrica: $A_{xy} = 1$ si los nodos están conectados o $A_{xy} = 0$ si no lo están. No lo usamos en nuestro trabajo porque se obtiene un resultado binario que no aporta una medida de similitud aceptable.

- **Common Neighbours o CN.** Cuanto más nodos vecinos tengan en común un par de nodos no conectados, más probabilidades tendrán de conectarse en un futuro. En esto se basa esta métrica, la cual determina el valor de similitud entre dos nodos como el número de nodos vecinos que tienen en común.

$$CN(x, y) = |N(x) \cap N(y)|$$

La versión con pesos sería la siguiente:

$$WCN(x, y) = \sum_{z \in N(x) \cap N(y)} A_{xz} + A_{yz}$$

- **Jaccard Neighbours o JN**. Se obtiene el valor de similitud de dos nodos como la comparación entre el número de vecinos en común que tienen esos nodos y el número total de vecinos que tienen ambos nodos.

$$JN(x, y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$$

Esta métrica no tiene versión con pesos.

- **Adar Adamic o AA**. Al igual que la métrica anterior, tiene en cuenta la suma del número de vecinos que tienen en común los nodos que se van a conectar. Sin embargo, esta métrica tiene en cuenta el conjunto de solapamiento más pequeño.

$$AA(x, y) = \sum_{z \in N(x) \cap N(y)} \frac{1}{\log|N(z)|}$$

La versión con pesos sería la siguiente:

$$WAA(x, y) = \sum_{z \in N(x) \cap N(y)} \frac{A_{xz} + A_{yz}}{\log(1 + WD(z))}$$

- **Preferential Attachment o PA**. Para calcular la probabilidad de que dos nodos se conecten, esta métrica tiene en cuenta que cuanto mayor número de enlaces tenga un nodo más probable será que se creen nuevos enlaces sobre él.

$$PA(x, y) = |N(x)| \cdot |N(y)|$$

Su correspondiente versión con pesos sería:

$$WPA(x, y) = WD(x) \cdot WD(y)$$

3.3.4. Métodos de agregación

En este Trabajo Fin de Máster se proponen cuatro métodos de agregación para calcular el valor de similitud agregado ag_j para un determinado problema p_j . De acuerdo a lo detallado en las secciones 3.3.1 y 3.3.2, partimos del siguiente estado: tenemos varios conjuntos $P_1..P_n$ de pares (p_i, s_i) , donde p_i es un problema y s_i es un valor de similitud (que se calcula de forma diferente dependiendo del grafo usado). Existen problemas que pueden repetirse en los distintos conjuntos $P_1..P_n$, con diferentes valores de similitud. Por tanto, necesitamos algún método de agregación como forma de obtener un único valor de similitud para cada problema repetido.

El **método de agregación basado en la similitud máxima** es el método de agregación propuesto más sencillo. El valor de similitud agregado ag_j para un problema p_j es igual al mayor valor de similitud s_j que tenga p_j en todas sus apariciones dentro de los conjuntos $P_1..P_n$. Si p_j aparece únicamente una vez, nos quedaremos con el único valor de similitud s_j que tiene.

Por otro lado, tenemos los **sistemas de votación** como métodos de agregación. La idea general de cómo funcionan los sistemas de votación está en conseguir tener un único valor de similitud final teniendo en cuenta cada aparición de un problema repetido en los distintos conjuntos de problemas. A continuación se detallan los tres sistemas de votación propuestos:

- **Votación simple.** El valor de similitud agregado ag_j es el número de veces que aparece p_j en los diferentes conjuntos $P_1..P_n$. Para el grafo de usuarios, ag_j va a ser el número de veces que aparece p_j en los conjuntos de problemas creados para cada usuario similar al usuario objetivo. En el caso del grafo de problemas, ag_j va a ser el número de veces que aparece p_j en los conjuntos de problemas similares a los realizados por el usuario objetivo.

- **Votación ponderada.** El valor de similitud agregado ag_j es la suma ponderada de los valores de similitud del problema en los distintos conjuntos.

$$ag_j = \sum_j \frac{s_j}{\sum_{i=1..n} s_i}$$

donde s_j es el valor de similitud de p_j en cada conjunto $P_1..P_n$.

- **Votación posicional.** En este caso el valor de similitud agregado ag_j depende del grafo que estemos utilizando. De forma general, ag_j se calcula de la siguiente forma:

$$ag_j = \sum \frac{1}{pos_j + 1}$$

Para el grafo de problemas, pos_j es la posición en la que aparece el problema p_j dentro de cada conjunto. En cada conjunto, los problemas deben ser ordenados previamente en orden decreciente por valor de similitud.

Para el grafo de usuarios, pos_j es la posición del usuario que aporta el problema p_j en la lista de usuarios similares al usuario objetivo. La lista de usuarios similares al usuario objetivo también debe ser previamente ordenada por su valor de similitud en orden decreciente.

Con el objetivo de aclarar cómo funcionan los sistemas de votación, se va a mostrar un ejemplo en el que se puede observar la importancia de la utilización de un sistema u otro para la obtención de diferentes resultados. Nuestro ejemplo (véase la Figura 3.6) está basado en el grafo de interacción de usuarios.

Partimos de cuatro usuarios, los cuales son similares al usuario objetivo u_t . Dependiendo de qué sistema de votación se utilice, obtendremos unos problemas recomendados u otros para u_t :

- Si elegimos $k = 1$ o $k = 2$, la lista de problemas recomendados a u_t sería la misma con todos los sistemas de votación: $[p_2]$ y $[p_2, p_1]$, respectivamente.

Usuario	pos_{u_j}	Problemas/Similitud	Valor de similitud
u_3	1	$\{(p_1, 5), (p_2, 5), (p_5, 5)\}$	$Sim_m(u_t, u_3) = 5$
u_1	2	$\{(p_1, 4), (p_2, 4), (p_4, 4)\}$	$Sim_m(u_t, u_1) = 4$
u_2	3	$\{(p_2, 2), (p_3, 2), (p_5, 2)\}$	$Sim_m(u_t, u_2) = 2$
u_4	4	$\{(p_1, 1), (p_2, 1), (p_3, 1)\}$	$Sim_m(u_t, u_4) = 1$

(a) Problemas y valores de similitud del usuario.

Votación simple	
p_1	$1 + 1 + 1 = 3$
p_2	$1 + 1 + 1 + 1 = 4$
p_3	$1 + 1 = 2$
p_4	1
p_5	$1 + 1 = 2$

$$L_r \text{ ordenada} = [p_2, p_1, p_3, p_5, p_4]$$

(b) Votación simple

Votación ponderada	
p_1	$5/12 + 4/12 + 1/12 = 0.83$
p_2	$5/12 + 4/12 + 2/12 + 1/12 = 1$
p_3	$2/12 + 1/12 = 0.25$
p_4	$4/12 = 0.33$
p_5	$5/12 + 2/12 = 0.58$

$$L_r \text{ ordenada} = [p_2, p_1, p_5, p_4, p_3]$$

(c) Votación ponderada

Votación posicional	
p_1	$1/1 + 1/2 + 1/4 = 1.75$
p_2	$1/1 + 1/2 + 1/3 + 1/4 = 2.08$
p_3	$1/3 + 1/4 = 0.58$
p_4	$1/2 = 0.5$
p_5	$1/1 + 1/3 = 1.33$

$$L_r \text{ ordenada} = [p_2, p_1, p_5, p_3, p_4]$$

(d) Votación posicional

Figura 3.6: Ejemplo de uso de los sistemas de votación

- Cuando $k = 3$, el resultado de la recomendación es $[p_2, p_1, p_3]$ para el sistema de votación simple y $[p_2, p_1, p_5]$ para la votación ponderada y posicional.

- Si escogemos $k = 4$ o $k = 5$, las listas de problemas recomendados calculadas usando cada uno de los sistemas de votación son diferentes. Por ejemplo, para $k = 4$, las listas resultantes serían $[p_2, p_1, p_3, p_5]$ para la votación simple; $[p_2, p_1, p_5, p_4]$, para la ponderada; y $[p_2, p_1, p_5, p_3]$, para la posicional.

3.4. Conclusiones

Como resumen de lo que se ha comentado en este capítulo, hemos destacado la necesidad de los jueces en línea de tener un sistema recomendador que ayude a sus usuarios a buscar problemas en la plataforma. Los usuarios de estas plataformas tienen una amplia oferta de ejercicios sobre la que poder trabajar, pero necesitan ayuda para poder determinar qué problemas les pueden interesar más. También se puede concluir que las técnicas de predicción de enlaces pueden proporcionarnos una forma de descubrir la similaridad que existe entre los usuarios y los problemas de un juez en línea, lo que, a su vez, puede proporcionarnos información para generar recomendaciones.

Teóricamente, los métodos aquí presentados pueden funcionar correctamente como recomendadores de jueces en línea, pero es necesario hacer una validación experimental de ellos. Además, sería interesante medir cuáles de los métodos propuestos se comportan mejor. Por ello, en el siguiente capítulo se va a realizar una evaluación de los modelos diseñados, ejemplificando su uso en el juez en línea Acepta el Reto.

Capítulo 4

Evaluación y análisis de resultados

En este capítulo se van a definir las tareas que se han realizado para hacer la evaluación de nuestros métodos recomendadores propuestos en el capítulo anterior. Primero se va a hacer un detallado estudio sobre cómo se evalúan los sistemas de recomendación (sección 4.1) y después, en base a este estudio, vamos a hacer la evaluación de nuestros métodos de recomendación. Para ello, se van a comentar los objetivos de hacer nuestra evaluación (sección 4.2), se va a hacer un análisis del conjunto de datos a utilizar en nuestros experimentos: la base de datos de Acepta el Reto (sección 4.3), después se pasará a mencionar las métricas de evaluación utilizadas (sección 4.4) y a comentar cómo se ha realizado el proceso de evaluación (sección 4.5) y por último, se mostrarán los resultados obtenidos (sección 4.6) y las conclusiones extraídas del análisis de los resultados (sección 4.7).

4.1. Evaluación de los sistemas de recomendación

Hacer una evaluación de los sistemas de recomendación y los algoritmos con los que se han implementado es una tarea de gran dificultad. Un mismo sistema de recomendación puede funcionar de diferentes formas dependiendo del conjunto de datos sobre el que se aplique. Algunos algoritmos funcionan mejor para conjuntos de datos en los que haya mayor número de usuarios que de productos, y viceversa. También depende de otras variables como la densidad de los envíos o de otras propiedades del conjunto de datos. Además, la dificultad

del proceso de evaluación depende del objetivo que tengamos para hacer esa evaluación. Algunos estudios muestran que cuando la evaluación se utiliza con el objetivo de apoyar las decisiones del sistema es más fácil medir cómo de correctamente funciona el sistema recomendador que si lo que se pretende es comprobar que el sistema aporta recomendaciones erróneas a los usuarios. Asimismo, hacer la elección de las métricas a utilizar para hacer una valoración de los resultados aportados por el sistema resulta bastante complicado [17]. Por tanto, es importante que, antes de hacer el proceso de evaluación, decidamos:

1. Cuáles serán los objetivos de los usuarios al utilizar el sistema recomendador a evaluar para así determinar si la plataforma que estamos evaluando se ajusta a las necesidades de los usuarios.
2. Qué conjunto de datos vamos a utilizar para hacer la evaluación y hacer un análisis del mismo.
3. Cuáles son las métricas de evaluación que podemos usar en nuestro problema y, de estas métricas, decidir cuáles serían las que más se adaptan a nuestras necesidades.
4. Cómo va a realizarse el proceso de evaluación.

4.1.1. Determinar los objetivos de los usuarios al utilizar el recomendador

Realizar una tarea tan compleja como la evaluación de un sistema recomendador debe comenzar por determinar cuáles son los objetivos del usuario cuando utiliza el software. Según Herlocker et al. [17] pueden existir múltiples razones para usar un sistema recomendador, pero hay algunos objetivos de carácter general, que serían los más comunes entre los usuarios de un sistema recomendador de cualquier tipo. Son los siguientes:

- **Ayuda en la toma de decisiones.** El usuario que utiliza un sistema recomendador quiere que el recomendador le ayude a tomar decisiones: qué libros leer, qué páginas

web visitar, etc.

- **Encontrar buenos productos.** Esta sería la principal tarea de un sistema recomendador: que el usuario encuentre el mejor producto según su punto de vista dentro del catálogo de posibilidades que tiene a su alcance.
- **Encontrar todos los buenos productos.** La mayoría de los recomendadores se centran en buscar los mejores productos para los usuarios, obviando que hay usuarios que necesitan encontrar todos los productos que les interesan o descartar todos aquellos productos que no les interesen.
- **Encontrar una buena secuencia de recomendación.** A veces es importante para los usuarios que la lista de productos recomendados tenga un orden específico, de forma que aquellos productos que puedan serle de más utilidad se encuentren en las primeras posiciones de la lista. Por ejemplo, cuando estamos leyendo sobre un tema específico, intentamos buscar aquellos libros o artículos que más se centren en ese tema. Lo mismo ocurre en las recomendaciones musicales: los usuarios intentan buscar canciones lo más acordes a sus gustos y lo más parecidas a canciones que ya escuchan.
- **Navegar por las recomendaciones.** Muchos usuarios navegan por las plataformas sin en realidad querer interactuar con la propia plataforma. Por ejemplo, en eBay, muchos usuarios ojean artículos, novedades u ofertas pero sin tener en realidad intención de comprar ninguno de esos artículos. En estos casos, la precisión del sistema recomendador es menos importante que la usabilidad de la propia plataforma.
- **Encontrar un recomendador fiable.** Muchos usuarios no se fían de los sistemas de recomendación y dudan de que estos sistemas vayan a ofrecerles productos que vayan a interesarles. Es importante tener esto en cuenta, ya que algunos de estos usuarios comprueban cuáles son las recomendaciones que les da el sistema simplemente por comprobar si acierta o no. Este problema es más un problema de confianza en las

nuevas tecnologías que en los sistemas de recomendación en sí, por lo que tendría que ser la propia plataforma la que despertara la confianza en los usuarios.

- **Mejorar su propio perfil.** Los sistemas de recomendación no suelen funcionar de forma efectiva cuando los usuarios son nuevos y no han interactuado con la plataforma y con los productos ofrecidos en ella (aunque depende del tipo de recomendador, como hemos visto en el Capítulo 2). Algunos de los usuarios de la plataforma tienen conciencia de ello, por lo que interactúan con la misma para mejorar su perfil y que el sistema recomendador empiece a actuar de forma adecuada y les ayude a encontrar los productos que les interesan.
- **Expresarse.** Muchos de los usuarios que utilizan este tipo de plataformas buscan expresar sus intereses y opiniones más que buscar nuevas recomendaciones. Por ejemplo, en la plataforma Goodreads¹ (red social de libros), además de evaluar los libros que han leído, los usuarios pueden hacer reseñas. La posibilidad de hacer reseñas es un aspecto muy importante para los usuarios de esta red social porque les permite dar su opinión con total libertad.
- **Ayudar a otros.** Muchos usuarios interactúan con la plataforma por ayudar a contribuir a la mejora del sistema de recomendación y así ayudar a otros. Estos usuarios, aunque no siempre, suelen tener también el objetivo de expresarse.
- **Influenciar a otros.** Unos pocos usuarios que utilizan sistemas de recomendación lo hacen por influenciar a otros, de forma que visitan o interactúan con unos determinados productos para así aumentar la posibilidad de que otros usuarios interactúen con esos mismos productos.

¹<https://www.goodreads.com/>

4.1.2. Determinar qué conjunto de datos utilizar

Determinar qué conjunto de datos utilizar es una de las tareas más importantes, ya que de ello van a depender los resultados que obtengamos en la evaluación. Siguiendo con el trabajo de Herlocker et al. [17] podemos destacar tres aspectos para decidir qué conjunto de datos utilizar para realizar la evaluación de un sistema de recomendación:

1. Decidir si la evaluación se realiza con un conjunto de datos ya existente (evaluación *offline*) o se realiza directamente con los usuarios (evaluación *online*) de la plataforma o incluso con una combinación de ambas. Las ventajas de la primera opción son muy obvias: se pueden hacer evaluaciones de grandes dimensiones de forma rápida y económica y con varios conjuntos de datos. La opción de hacer la evaluación directamente con los usuarios, de forma que ellos trabajen directamente con el software y observen las recomendaciones que les proporcione, es una alternativa más costosa en cuanto a recursos humanos a organizar, pero también mucho más efectiva y fiable en cuanto a que podremos saber de primera mano la opinión de los usuarios con respecto a las recomendaciones que se le proporcionan.
2. Decidir si podemos utilizar un conjunto de datos simulado en el caso de no tener uno real. En general el uso de un conjunto de datos real siempre va a ser mejor que uno simulado (uno creado por nosotros para hacer la evaluación), ya que siempre se va a ajustar mejor a la realidad y obtendremos resultados más efectivos. Los conjuntos de datos simulados se deben usar en contadas ocasiones cuando no dispongamos de conjuntos de datos reales, pero sólo como una primera toma de contacto con el sistema recomendador hasta tener un conjunto de datos real.
3. Las propiedades del conjunto de datos. Las propiedades se pueden dividir en tres grupos: las características de dominio, que son las que muestran el carácter de los productos a recomendar; las características inherentes, que muestran la naturaleza del sistema recomendador; y las características de las muestras, que reflejan la distribución

de los datos del conjunto de datos. Las características del dominio incluyen el tema del contenido a recomendar, las tareas que pueden realizar los usuarios con el sistema, las preferencias de los usuarios, la novedad de los productos a recomendar y la calidad de los mismos. Por otro lado, las características inherentes comprenden la escala de la valoración de los productos, si las interacciones son implícitas (las interacciones directas del usuario con la plataforma) o explícitas (obtenidas a partir del comportamiento del usuario), la presencia o ausencia de fecha de interacción (*timestamp*) entre otras características. Por último, las características de muestra incluyen la distribución de las diferentes propiedades en el conjunto de los datos, el tamaño del conjunto de datos o la densidad de las interacciones dentro del conjunto de datos.

Cuando se haya decidido qué conjunto de datos utilizar en nuestra evaluación y se haya hecho un análisis del mismo, se pasará a decidir con qué métricas se van a valorar los resultados y cómo se va a hacer este proceso.

4.1.3. Métricas a utilizar durante la evaluación

Las métricas de evaluación se utilizan como técnicas para determinar cómo de similares son una lista de recomendación de productos para un usuario concreto ofrecida por un sistema de recomendación y la lista de productos en orden de preferencia que tendría ese usuario en la realidad. La lista de métricas de evaluación es bastante amplia, por lo que, a veces, puede resultar complicado elegir cuál o cuáles de estas métricas son las más adecuadas para la evaluación que queramos realizar. Este conjunto de métricas se podrían dividir en tres clases: métricas de precisión en la predicción, métricas de precisión en la clasificación y métricas de precisión en el orden u orden [\[17\]](#).

Las métricas de precisión en la predicción son aquellas que miden cómo de similares son las calificaciones de los productos por parte de los usuarios. Estas métricas están orien-

tadas a evaluar aquellos sistemas de recomendación cuyos usuarios usen este sistema por su necesidad de tomar decisiones a la hora de elegir nuevos productos con los que interactuar.

Las de precisión en la clasificación, sin embargo, son adecuadas cuando el objetivo principal de los usuarios que utilizan el sistema recomendador es encontrar los productos buenos para ese usuario. Esto suele ocurrir cuando los usuarios tienen preferencias binarias (por ejemplo, si les gusta o no les gusta un producto). Lo que hacen estas métricas es medir cómo de bien se comportan los sistemas de recomendación cuando deciden si un producto es bueno o malo para un determinado usuario.

Por último, las métricas de precisión en el orden son las métricas que van a medir cómo de bien ordenadas están las listas de productos que genera el sistema para un usuario desde el punto de vista del usuario al que se le recomienda.

4.1.4. Proceso de evaluación de los sistemas recomendadores

Los procesos de evaluación sobre sistemas recomendadores son procesos iterativos sobre un conjunto de datos, que se inspiran en métodos de evaluación de sistemas de Recuperación de Información. Es una de las etapas más importante en la evaluación del sistema recomendador porque se va a definir cómo se va a realizar la evaluación en función de los datos obtenidos en las anteriores etapas: cómo se va a particionar el conjunto de datos, cuántas repeticiones se van a realizar, cómo se van a utilizar las métricas de evaluación y cómo se van a agregar los resultados de las iteraciones para obtener un resultado final.

Los procesos de evaluación tienen tres etapas principales:

- Partición del conjuntos de dato en dos conjuntos: el conjunto de entrenamiento para generar las recomendaciones y el conjunto de evaluación para hacer la evaluación. La

partición se puede realizar de diferentes formas, por ejemplo por el *timestamp* de la interacción usuario-producto.

- Generación de las recomendaciones a partir del conjunto de entrenamiento. Se generan listas de recomendación de productos para los usuarios seleccionados gracias a la información contenida en el conjunto de entrenamiento.
- Comparación de las listas de recomendación generadas en el paso anterior con información contenida en el conjunto de evaluación. Para ello se utilizan las métricas de evaluación.

Algunos procesos habitualmente usados en la evaluación de los sistemas de recomendación son:

- ***Holdout***. Durante un número fijo de iteraciones se divide el conjunto de datos en un conjunto de entrenamiento y un conjunto de evaluación distintos [2].
- ***Crossfold***. El proceso de evaluación se divide en varias etapas, en las que siempre vamos a partir de un conjunto de datos previamente dividido en subconjuntos. En cada una de las etapas vamos a escoger uno de esos subconjuntos como el de evaluación y el resto se van a unir para formar el conjunto de entrenamiento. En cada etapa se va a escoger un subconjunto diferente como conjunto de evaluación [2].
- ***Leave-one-out***. Es equivalente al *crossfold* pero el tamaño de conjunto de evaluación es 1, mientras que el de entrenamiento es $Q - 1$, donde Q es el tamaño del conjunto de datos original [2].

También se pueden crear procesos específicos para el experimento al que se aplique. Por ejemplo, el trabajo de Said et al. [42]: un protocolo que evalúa recomendaciones de uno o varios productos a un usuario final, teniendo en cuenta especialmente las características específicas del usuario al que se hace las recomendaciones. O también las diferentes técnicas

aplicadas en el trabajo de Bellogín et al., donde se especifican cinco formas de procesos de evaluación para sistemas de recomendación [6].

Habiendo estudiado cómo hacer de forma genérica la evaluación de los sistemas de recomendación, en las siguientes secciones vamos a detallar cómo se va a realizar la evaluación de nuestros métodos de recomendación propuestos con el conjunto de datos del juez en línea Acepta el Reto (en adelante ACR). A continuación se especifican brevemente los objetivos de la evaluación realizada con ACR, se analiza el conjunto de datos de este juez, se determinan qué métricas de evaluación hemos usado y por último, se muestra el proceso seguido para realización de los experimentos.

4.2. Objetivos de la evaluación realizada

Los jueces en línea ofrecen una gran cantidad de ejercicios que sus usuarios pueden resolver y a cuyas soluciones puede proporcionar un veredicto. Sin embargo, los usuarios, ante la amplia oferta de ejercicios que presenta la plataforma, no saben decidir qué problemas son lo que más les puede interesar realizar. Por tanto, al utilizar el sistema recomendador de un juez en línea a los usuarios les gustaría tener una herramienta que les oriente y ayude a la hora de encontrar ejercicios en la plataforma. Teniendo esto en cuenta y, en términos de lo comentado en la sección 4.1.1, se puede decir que los principales objetivos de nuestros usuarios son encontrar buenos productos (problemas) y tener ayuda en la toma de decisiones. También se puede decir que encontrar una buena secuencia de recomendación sería otro objetivo que podrían tener los usuarios de un juez en línea.

Según lo anterior, el objetivo general de la realización de nuestra evaluación es comprobar que los métodos recomendadores propuestos en el capítulo anterior generan listas de recomendación adecuadas para cada usuario. Con ello, se cumplirían además las expectati-

vas de los futuros usuarios, aportándoles satisfacción a la vez que ayuda. También tenemos intención de averiguar cuáles de los métodos recomendadores son más eficaces en su tarea de sugerir problemas a los estudiantes para así hacer una propuesta más sólida y fundamentada. Por último, tenemos como objetivo realizar un análisis comparativo de los métodos propuestos teniendo en cuenta los ejes de variabilidad de nuestros modelos (el grafo construido, la métrica de similitud escogida o el tamaño de la lista de problemas a recomendar).

4.3. Conjunto de datos: Acepta el Reto

En nuestro caso, el conjunto de datos a utilizar para nuestra evaluación es la base de datos de ACR, el juez en línea de nuestra facultad. A continuación se van a comentar algunas características relevantes de ACR y se va a realizar un análisis de la información contenida en su base de datos. Parte del contenido que se va a detallar en el resto del capítulo está publicado en los artículos [10] y [21].

Acepta el Reto es un juez en línea desarrollado en 2014 por Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín, profesores de la Facultad de Informática de la Universidad Complutense de Madrid (UCM). El objetivo inicial de la plataforma era ayudar a los alumnos de la facultad a superar algunas de sus asignaturas, como la de Estructuras de Datos y Algoritmos. La plataforma ha crecido tanto que actualmente se celebran con ella concursos de programación como el concurso *AdaByron*², concurso universitario a nivel de la Comunidad de Madrid, *ProgramaMe*³, concurso para alumnos de ciclos formativos o *las 12 UVas*⁴, el cual se celebra todos los años el 31 de diciembre.

La base de datos de ACR contiene la información de todas las *submissions* o entregas de

²<http://ada-byron.es/2017/>

³<http://programa-me.com/2017/reg/>

⁴<http://las12uvas.es>

soluciones a problemas realizadas en el juez en línea. Desde el punto de vista del sistema, una entrega de un ejercicio se puede interpretar como una tupla (d, p, u, c, v) , donde d es la fecha de entrega de la solución al ejercicio, p es el identificador del problema resuelto, u es el identificador del usuario que ha enviado la solución para el problema p , c es el código enviado por el usuario u para el problema p y v el veredicto otorgado por el juez al usuario en función de la solución enviada.

Como se ha comentado en el capítulo anterior, cada juez en línea tiene su propio código de valoración de los ejercicios resueltos. En el caso de ACR, los posibles veredictos que puede otorgar a un usuario por la solución que ha entregado para resolver un determinado ejercicio son:

- ***Accepted* o AC.** La solución proporcionada por el estudiante es la misma que la solución del juez en línea. Además, la solución no supera las restricciones de tiempo y memoria que limita el juez.
- ***Presentation Error* o PE.** La solución entregada es correcta pero el formato del *output* no es exactamente igual al que tiene la solución del juez. Puede diferir en espacios en blanco o saltos de línea.
- ***Wrong Answer* o WA.** El programa entregado por el estudiante falla cuando intenta escribir una o más de las posibles soluciones del juez.
- ***Compile Error* o CE.** El código del estudiante no compila.
- ***Runtime Error* o RE.** El programa falla durante su ejecución (por fallo de segmentación, excepción de punto flotante, etc.).
- ***Memory Limit Exceeded* o MLE.** El programa consume demasiada memoria.
- ***Output Limit Exceeded* u OLE.** El programa del estudiante escribe un *output* demasiado largo. Esto suele ocurrir porque el programa contiene un bucle infinito.

- ***Time Limit Exceeded* o TLE.** La ejecución del programa tarda demasiado y el límite de tiempo del juez se excede.

Cabe destacar que la base de datos de ACR va a contener mucha información repetida, ya que es redundante e irrelevante para nuestro objetivo conocer cuántas veces ha interactuado un usuario con el mismo problema: lo único que nos interesa es que un determinado usuario u ha interactuado con un determinado problema p . Por tanto, vamos a hacer un filtro de la base de datos de ACR para generar las recomendaciones: sólo se va a tener en cuenta la última entrega resuelta por el usuario u para un problema p . Si no lo ha resuelto, se tendrá en cuenta la última entrega intentada. Es por esto por lo que la matriz de adyacencia construida para implementar los recomendadores propuestos para ACR va a ser definida en función de si el usuario ha intentado resolver el problema o no. Es decir, la matriz $A = a_{ij}$ tendrá $a_{ij} = 1$ cuando el usuario u_i ha intentado resolver el problema p_j (o lo ha resuelto) y $a_{ij} = 0$ en caso contrario.

4.3.1. Análisis de Acepta el Reto

Antes de evaluar los métodos de recomendación diseñados, tenemos que hacer un análisis de las interacciones realizadas por los usuarios de la plataforma, es decir, de las tuplas almacenadas en el juez y de toda la información que se puede extraer de esas interacciones. El objetivo de hacer este análisis es entender el conjunto de datos para saber a qué nos vamos a enfrentar en la evaluación. El análisis de la base de datos se va a realizar sobre la base de datos original y sobre la base de datos filtrada, en la que sólo se tienen en cuenta las últimas entregas intentadas o resueltas.

El análisis de la base de datos de ACR se muestra en la Tabla 4.1. Se ha realizado este tipo de análisis siguiendo el modelo presentado en el artículo de Dooms et al. [13], don-

Tabla 4.1: Análisis de la base de datos de Acepta el Reto.

Métrica	<i>Original</i>	<i>Filtrada</i>
# Entregas	110.364	25.151
# Problemas	289	289
# Usuarios	3.678	3.678
Densidad	0,10	0,02
Fecha de primera entrega	17/02/2014	17/02/2014
Fecha de última entrega	13/02/2017	13/02/2017
Tiempo de diferencia	1092 días	1092 días
Problemas		
Máximo # entregas por problema	5.613	1.290
Mediana # entregas por problema	232	52
Media # entregas por problema	381,88	87,03
Mínimo # entregas por problema	8	3
# Problemas con al menos 10 entregas	276	253
# Problemas con al menos 50 entregas	216	138
# Problemas con al menos 100 entregas	146	56
Usuarios		
Máximo # entregas por usuario	2.576	250
Mediana # entregas por usuario	10	3
Media # entregas por usuario	30,01	6,84
Mínimo # entregas por usuario	1	1
# Usuarios con al menos 5 entregas	2.415	1.160
# Usuarios con al menos 10 entregas	1.790	606
# Usuarios con al menos 20 entregas	1.198	220
Veredictos		
# Entregas con AC-PE	36.824	18.067
# Entregas con CE	7.061	465
# Entregas con RLE	31.924	1.803
# Entregas con WA	33.443	3.146

de se hace una comparativa del conjunto de datos de MovieTweetings⁵ con el de MovieLens⁶.

En la Tabla 4.1 se pueden observar las características generales del contenido de la base de datos de ACR. Se pueden observar tanto para la base de datos original, como para la filtrada los siguientes datos:

⁵<https://github.com/sidooms/MovieTweetings>

⁶<https://movielens.org/>

- El número de entregas, problemas y usuarios de ACR.
- La densidad del conjunto de datos, que se calcula con la fórmula [13]:

$$Densidad = \frac{\#Entregas}{\#PosiblesEntregas} = \frac{\#Entregas}{\#Usuarios \times \#Problemas}$$

- Las fechas del primer y último ejercicio subido a ACR, así como el tiempo de diferencia entre estas fechas.
- El máximo, el mínimo, la media y la mediana del número de entregas por problema y por usuario.
- El número de problemas y usuarios con al menos un determinado número de entregas (10, 50 y 100 entregas como mínimo para los problemas y 5, 10 y 20 entregas como mínimo para los usuarios).
- El número de entregas según los diferentes veredictos que puede otorgar ACR. El número de veredictos con AC y PE se calcula en conjunto porque las soluciones subidas por los usuarios se pueden considerar como correctas en ambos casos.

Analizando los datos mostrados en la tabla, se puede destacar la gran diferencia de resultados obtenidos entre la base de datos original y la filtrada. El número de entregas se reduce de 110.364 a 25.151, así como todos los datos relacionados con el número de entregas en el conjunto de datos: máximo/mínimo/mediana/media número de entregas por problema o por usuario, la densidad, los veredictos, etc. El descenso de cifras es muy alto, de lo que se puede decir que el filtrado realizado sobre el conjunto de datos es adecuado ya que hemos eliminado gran cantidad de información redundante. También cabe destacar que el número de usuarios y el número de problemas no cambian en la base de datos original y en la filtrada. Ningún usuario se elimina porque todos han realizado intentos en el juez. Ocurre lo mismo con el número de problemas.

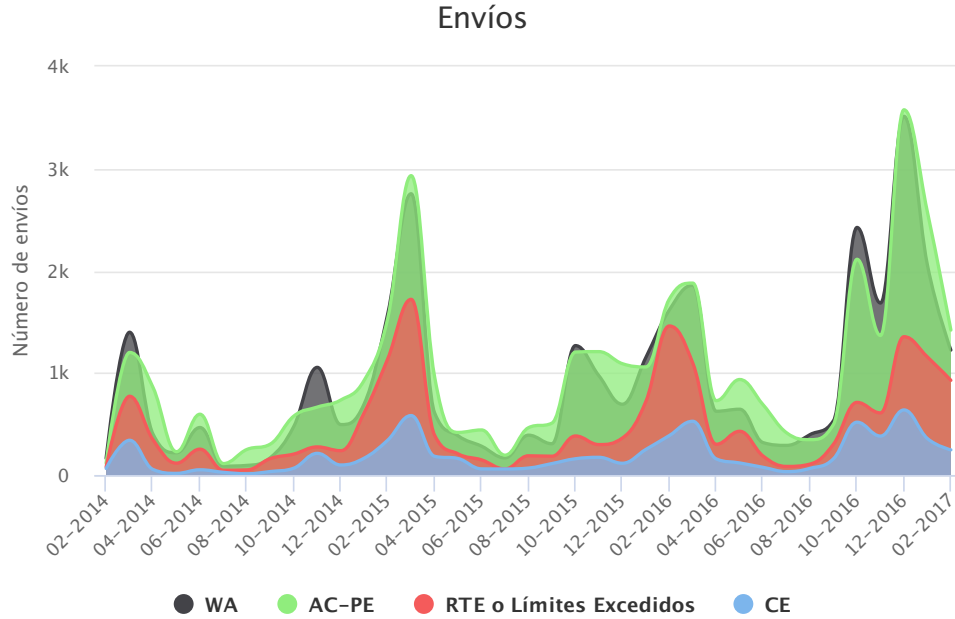


Figura 4.1: Gráfico de envíos en Acepta el Reto entre febrero de 2014 y febrero de 2017, categorizados por veredicto.

Por otro lado, la densidad es un aspecto importante a tener en cuenta en nuestro estudio. Es alta en la base de datos original en comparación con la obtenida al filtrar el conjunto de datos original. Esto indica de nuevo que hemos eliminado gran cantidad de información irrelevante.

Durante el trabajo de análisis se ha realizado también un estudio temporal sobre el número de entregas realizadas en ACR desde que empezó a utilizarse en febrero de 2014, hasta el momento en el que comenzamos a realizar la evaluación de nuestros recomendadores en febrero de 2017. En las Figuras 4.1 y 4.2 se puede observar respectivamente el número de envíos realizados en ACR en ese intervalo de tiempo, y el número acumulado de envíos también entre esas fechas.

En la primera gráfica, destacan ciertos momentos en los que el número de envíos es mucho mayor que en el resto del tiempo. Estos puntos son las fechas en las que se han

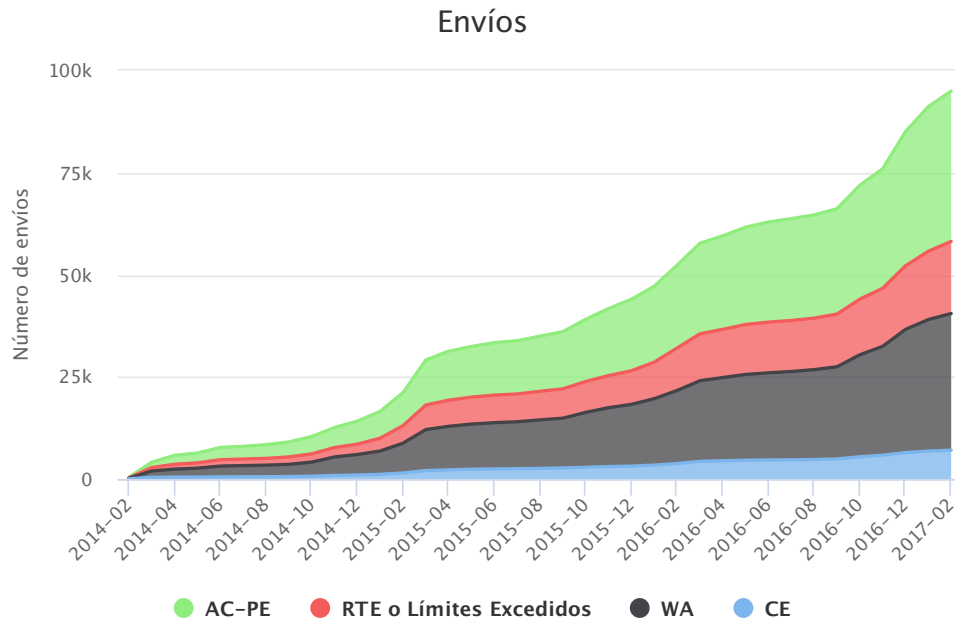


Figura 4.2: Gráfico acumulado de envíos en Acepta el Reto entre febrero de 2014 y febrero de 2017, categorizados por veredicto.

celebrado concursos de programación en la plataforma. Si nos fijamos en estos puntos, se puede afirmar que cada vez hay más concursos celebrados con ACR, ya que hay más picos y de mayor altura en 2016 que en 2014. También es importante destacar que, en los intervalos de tiempo entre concursos, ha aumentado el número de entregas. Es decir, ACR se está haciendo cada vez más usado y popular, tanto por la celebración de los concursos como por su uso para estudiar programación.

En la segunda gráfica, se corrobora lo mismo que en la anterior: el número de entregas se ha incrementado rápidamente en el último año, dando lugar a una base de datos con información de gran cantidad de interacciones.

4.4. Métricas de evaluación

En nuestros experimentos hemos utilizado algunas de las métricas de evaluación más conocidas. Son las siguientes:

- *Precision, Recall, F-score*. Son métricas de tipo precisión en clasificación. *Precision* calcula la probabilidad de que un producto seleccionado por el sistema recomendador sea relevante para el usuario. Por el contrario, *Recall* mide la probabilidad de que un producto relevante para el usuario sea escogido por el sistema recomendador. *F-score* une ambos conceptos en una misma métrica.

$$P = \frac{N_{rs}}{N_s} \qquad R = \frac{N_{rs}}{N_r} \qquad F_1 = \frac{2*P*R}{P+R}$$

donde P es el valor que toma *Precision*, R , el que toma *Recall*, F_1 , el de *F-score*, N_{rs} es el número de productos relevantes seleccionados, N_s , el número de productos seleccionados y N_r , el número de productos relevantes.

Los productos relevantes en nuestro caso son los problemas que ha intentado resolver o ha resuelto el usuario en el conjunto de evaluación.

- *At least one hit* (1-hit). Ratio de recomendaciones en las que al menos un problema de los recomendados fue intentado por el usuario en el conjunto de evaluación. Pertenece al grupo de métricas de precisión en la clasificación y se integra dentro de las métricas *Success@k*, que indican si se cumple una condición de éxito en una lista de recomendaciones.
- *Mean Reciprocal Rank* (MRR). Evalúa la calidad de la lista de recomendaciones a través de la posición del primer elemento en la lista que sea relevante (que haya realizado el usuario en el conjunto de evaluación). Pertenece al grupo de métricas de precisión en el orden.

$$MRR = \frac{1}{rank_i}$$

donde MRR es el valor que toma la métrica y $rank_i$ es la posición del primer elemento intentado por el usuario que aparece dentro de la lista de recomendación.

4.5. Proceso de experimentación

Para poder evaluar los modelos de recomendación diseñados, se van a realizar experimentos implementando los métodos propuestos utilizando la base de datos de Acepta el Reto. Lo que se pretende es obtener listas de problemas recomendados para los usuarios del juez en línea y analizar las listas de problemas obtenidas con el fin de determinar si esos problemas se pueden considerar como interesantes para el usuario al que se han recomendado.

Para realizar la evaluación de cada uno de los sistemas de recomendación implementados, lo primero que hay que hacer es generar dos conjuntos a partir del conjunto de datos filtrado:

- El conjunto de entrenamiento, que comprenderá el conjunto de entregas realizadas antes de la fecha umbral t y que se utilizará como base para generar las recomendaciones. Este conjunto se va a filtrar para que sólo se tengan en cuenta problemas resueltos (aquellas entregas con PE o AC como veredicto), de forma que los problemas que los usuarios ya han resuelto no se les vuelvan a recomendar.
- El conjunto de evaluación, que comprenderá las acciones realizadas después de la fecha t , y que se va a usar para evaluar las recomendaciones generadas en el sistema. El conjunto de evaluación, sin embargo, se mantendrá con el último intento por cada usuario-problema.

El siguiente paso es decidir a qué conjunto de usuarios queremos hacer las recomendaciones. En el conjunto de datos original se encuentran todas las interacciones realizadas en

el sistema, tanto las realizadas por usuarios expertos (aquellos usuarios que utilizan asiduamente el sistema), como por usuarios puntuales (aquellos usuarios que han utilizado apenas unas pocas veces el sistema). En la evaluación no interesa incluir a los usuarios puntuales, ya que con tan poca información no se puede deducir o evaluar cómo se comportarían dichos usuarios. Para ello, hay que comprobar cuántos problemas ha hecho cada usuario antes y después de la fecha umbral, de forma que sólo tengamos en cuenta a aquellos usuarios que hayan realizado como mínimo un cierto número de problemas en ambos períodos de tiempo. Los usuarios de esa lista serán el usuario objetivo de nuestros recomendadores (lista que denotaremos como U_o , en adelante). En nuestros experimentos, consideramos que los usuarios expertos son aquellos que han realizado al menos 5 problemas antes y después de t .

Durante el inicio de este TFM, se hizo un estudio del trabajo antecedente realizado el año pasado [21]. En este trabajo se usaba como fecha umbral el 30/06/2015, ya que era la fecha en la cual se había realizado la mitad de las entregas que se tenían almacenadas en ese momento en el juez. Como se comenzó haciendo una réplica de este trabajo, utilizando incluso el mismo conjunto de datos, se decidió incorporar esta fecha como t en nuestros primeros experimentos. Sin embargo, tras un análisis del conjunto de datos finalmente utilizado, se eligió como fecha umbral t para dividir el conjunto de datos original el 20/10/2016. Se eligió esta fecha porque es con la que se obtuvo un mayor número de usuarios que habían intentado al menos 5 problemas antes y después de la misma. El número de usuarios obtenido fue 65.

Hecho esto, el siguiente paso es implementar los métodos de recomendación propuestos en el capítulo anterior para recomendar listas de problemas a nuestros usuarios objetivo. Cada método de recomendación va a funcionar a partir del grafo de interacción construido con la información del conjunto de entrenamiento. Para $t = 20/10/2016$, los grafos creados tienen las siguientes características:

- El grafo con nodos-problemas tiene 169 nodos y 14.149 enlaces y una densidad de

49,5 %.

- El grafo con nodos-usuarios es más grande pero tiene menor densidad. Está formado por 2.007 nodos, 624.206 enlaces y su densidad es de 15,5 %.

Debido a la alta densidad y cantidad de enlaces que tienen nuestros grafos, se decidió hacer un filtrado de sus enlaces, eliminando todos aquellos enlaces del grafo que no tengan como mínimo un peso igual a 5. Los grafos mantienen su número de nodos, pero los enlaces y la densidad se reducen:

- El grafo con nodos-problemas tiene ahora 10.837 enlaces y una densidad de 37,9 %.
- El grafo con nodos-usuarios tiene 16.805 enlaces y densidad de 0,4 %.

Una vez construido el grafo, ya sea el de usuarios o problemas, se ha implementado un recomendador por cada una de las métricas de similitud de predicción de enlaces y cada uno de los métodos de agregación vistos en el capítulo anterior. Para cada usuario en U_o se han generado recomendaciones de tamaño k . En nuestro trabajo, hemos implementado recomendaciones con $k \in [1.,10]$.

Después de esto, necesitamos saber si cada recomendador ha sabido predecir cuáles son los problemas que más podían interesar a nuestros usuarios objetivo. Para ello, comparamos las listas de recomendaciones generadas por el recomendador con los problemas realizados por los usuarios objetivo después de la fecha umbral t , que se encuentran en el conjunto de evaluación. Estas comparaciones se realizarán con las métricas mencionadas en la sección anterior [4.4](#).

En total, se han implementado 64 experimentos a partir de todas las posibles combinaciones de los recomendadores diseñados sobre el conjunto de datos de ACR. Se han utilizado:

- Dos grafos de interacción: uno basado en la interacción de problemas y otro en la interacción de usuarios.
- Ocho métricas de similitud: AA, CN, EW, JN, PA, WCN, WAA y WPA (comentadas en el capítulo anterior 3.3.3).
- Cuatro métodos de agregación: el método de agregación basado en la similitud máxima y los tres sistemas de votación (también todos ellos detallados en el anterior capítulo 3.3.4).

La implementación de los experimentos se ha realizado con el lenguaje Python (versión 3.4.0), elegido por la gran cantidad de librerías que posee para manejo y análisis de datos (como la librería Pandas ⁷), así como de grafos (como la librería NetworkX ⁸). Como entorno de desarrollo se han utilizado los *notebooks* interactivos de Anaconda⁹, los cuáles son muy útiles para la realización de los experimentos. El código implementado y todas las tablas y gráficas generadas se encuentran disponibles en GitHub¹⁰.

Con los 64 sistemas de recomendación, los diferentes tamaños de listas de recomendación (k) aplicados durante los experimentos, la fecha umbral utilizada para hacer la evaluación y las 5 métricas de evaluación propuestas, generamos durante esta etapa una inmensa cantidad de resultados, que se comentarán y analizarán a continuación.

4.6. Análisis de los resultados

El objetivo de hacer el análisis de los resultados obtenidos en la evaluación es determinar qué métodos utilizados en nuestros experimentos se comportan mejor sobre el juez en línea

⁷<http://pandas.pydata.org/>

⁸<https://networkx.github.io/>

⁹<https://docs.continuum.io/anaconda/>

¹⁰<https://github.com/martcaro/TrabajoFinDeMaster>

Acepta el Reto. Para ello se van a comparar los datos dependiendo de:

- Los diferentes grafos utilizados: usuarios y problemas.
- Los diferentes métodos de agregación utilizados. Se van a diferenciar los resultados dependiendo del grafo utilizado.
- Los diferentes tamaños de las listas de recomendación a generar.

En todas las secciones se hace una comparativa de los resultados obtenidos con las distintas métricas de similitud y las diferentes métricas de evaluación utilizadas.

Como muestra, en este capítulo se van a presentar los datos referentes a $k = 5$ y $t = 20/10/2016$, de forma que se pueda tener una visión general de lo que se ha elaborado. Se ha elegido ese valor en concreto por ser una muestra representativa de lo que se ha obtenido en la evaluación. El resto de datos se encuentran disponibles en GitHub¹¹.

4.6.1. Análisis según el grafo de interacciones

Si comparamos los datos obtenidos en función del grafo de interacciones, se puede observar que los resultados dependen de las métricas de similitud utilizadas. Como se puede contemplar en la Tabla 4.2, tabla que hace referencia a los datos obtenidos para el grafo de problemas y el grafo de usuarios usando el sistema de agregación basado en la similitud máxima, los valores conseguidos dependen de las métricas de similitud utilizadas. Con el grafo de problemas como base se obtienen mejores resultados si aplicamos las métricas de similitud con pesos: EW, WCN, WAA y WPA. Por el contrario, obtenemos mejores resultados utilizando el de usuarios si generamos las recomendaciones con las métricas de similitud sin pesos. La única excepción a esta regla correspondería a los resultados obtenidos con la métrica PA. Los datos obtenidos con PA son mejores sobre el grafo de usuarios

¹¹<https://github.com/martcaro/TrabajoFinDeMaster>

Tabla 4.2: Resultados de la evaluación con las diferentes métricas de similitud con base en el grafo de usuarios y de problemas para listas de recomendación de $k = 5$ y $t = 20/10/2016$. Los mejores resultados obtenidos entre el grafo de usuarios y el de problemas está marcado en negrita. El mejor valor obtenido por métricas está marcados con *.

Métrica	Grafo	Precision	Recall	F1	MRR	1-hit
CN	Prob-Prob	0.111	0.055	0.070	0.194	0.400
	User-User	0.182	0.100	0.122	0.398	0.523
AA	Prob-Prob	0.129	0.063	0.081	0.241	0.446
	User-User	0.148	0.076	0.094	0.388	0.462
JN	Prob-Prob	0.098	0.044	0.057	0.185	0.323
	User-User	0.123	0.057	0.073	0.318	0.400
PA	Prob-Prob	0.126	0.061	0.078	0.329	0.462
	User-User	0.123	0.057	0.073	0.318	0.400
EW	Prob-Prob	0.268*	0.136	0.172*	0.486*	0.677
	User-User	0.175	0.092	0.114	0.344	0.462
WCN	Prob-Prob	0.252	0.132	0.165	0.421	0.785*
	User-User	0.114	0.051	0.065	0.309	0.369
WAA	Prob-Prob	0.249	0.131	0.164	0.423	0.785*
	User-User	0.114	0.051	0.065	0.309	0.369
WPA	Prob-Prob	0.265	0.137*	0.172*	0.428	0.785*
	User-User	0.123	0.057	0.073	0.318	0.400

que sobre el de problemas, a pesar de ser una métrica sin peso. Haciendo un visionado global son EW (con *Precision*, *F1* y *MRR*) y WPA (con *Recall*, *F1* de nuevo y *One hit* –esta última junto con WCN y WAA–) las métricas con las que se obtienen los mejores resultados.

4.6.2. Análisis según el método de agregación

En esta sección, vamos a hacer una comparativa de los resultados obtenidos con los diferentes métodos de agregación, dividiendo el análisis dependiendo del grafo utilizado. Primero vamos a hacer el análisis basándonos en los resultados obtenidos con el grafo de interacción de usuarios y después con el de problemas.

Si nos fijamos en los resultados mostrados en la Tabla 4.3, donde se muestran los resultados obtenidos con los diferentes métodos de agregación utilizando como base el grafo de

usuarios, podemos observar que cualquiera de los sistemas de votación utilizados para este grafo proporciona mejores datos que cualquiera de los proporcionados por el método de agregación basado en la similitud máxima. Si comparamos los resultados con los obtenidos en base al grafo de problemas, se observa de nuevo que cualquiera de los valores conseguidos con los sistemas de votación para el grafo de usuarios es mejor que cualquiera de los valores conseguidos con el grafo de problemas (independientemente del método de agregación usado). El sistema de votación posicional funciona, en general, mejor en combinación con las métricas sin pesos, y el sistema de votación ponderada, con las métricas de similitud con pesos. Esto se debe a que los sistemas de votación posicional eliminan las diferencias de similitud entre los problemas de una lista, por lo que en realidad se tiene en cuenta la posición de los problemas en la lista y no los propios pesos de los enlaces. Por tanto, los datos obtenidos se ajustan a lo que podríamos esperar. La excepción son los valores obtenidos con PA y WPA, como ya se observó en la sección anterior. Con la métrica PA se obtienen mejores resultados con la votación ponderada y simple, mientras que con la WPA, la métrica con resultados prometedores es la posicional. En general, la métrica CN (utilizando el sistema de votación posicional) es la que obtiene los mejores datos de toda la tabla analizada para las métricas de evaluación *Precision*, *Recall* y *F1*. Para MRR, la mejor métrica es EW (y el sistema de votación posicional), mientras que para *One hit* es AA con sistema de votación posicional.

Tabla 4.3: Resultados de la evaluación del sistema basado en el grafo de usuarios: se muestran los resultados obtenidos con y sin sistemas de votación para $k = 5$ y $t = 20/10/2016$. Los mejores resultados obtenidos entre el grafo de usuarios y el de problemas está marcado en negrita. El mejor valor obtenido por métricas está marcados con *.

Métrica	Agregación	Precision	Recall	F1	MRR	1-hit
CN	Similitud	0.182	0.100	0.122	0.398	0.523
	Simple	0.274	0.142	0.179	0.472	0.754
	Ponderada	0.314	0.166	0.208	0.474	0.831
	Posicional	0.357*	0.192*	0.239*	0.588	0.831
AA	Similitud	0.148	0.076	0.094	0.388	0.462
	Simple	0.274	0.142	0.179	0.472	0.754
	Ponderada	0.323	0.170	0.214	0.482	0.831
	Posicional	0.351	0.189	0.235	0.567	0.877*
JN	Similitud	0.123	0.057	0.073	0.318	0.400
	Simple	0.274	0.142	0.179	0.472	0.754
	Ponderada	0.274	0.143	0.181	0.442	0.754
	Posicional	0.348	0.186	0.232	0.589	0.800
PA	Similitud	0.123	0.057	0.073	0.318	0.400
	Simple	0.274	0.142	0.179	0.472	0.754
	Ponderada	0.274	0.143	0.181	0.442	0.754
	Posicional	0.252	0.130	0.164	0.409	0.754
EW	Similitud	0.175	0.092	0.114	0.344	0.462
	Simple	0.345	0.187	0.233	0.587	0.815
	Ponderada	0.348	0.187	0.234	0.598	0.846
	Posicional	0.342	0.185	0.230	0.602*	0.800
WCN	Similitud	0.114	0.051	0.065	0.309	0.369
	Simple	0.274	0.142	0.179	0.472	0.754
	Ponderada	0.289	0.152	0.191	0.488	0.800
	Posicional	0.271	0.144	0.180	0.439	0.785
WAA	Similitud	0.114	0.051	0.065	0.309	0.369
	Simple	0.274	0.142	0.179	0.472	0.754
	Ponderada	0.317	0.168	0.211	0.494	0.831
	Posicional	0.283	0.147	0.186	0.436	0.800
WPA	Similitud	0.123	0.057	0.073	0.318	0.400
	Simple	0.274	0.142	0.179	0.472	0.754
	Ponderada	0.280	0.146	0.184	0.450	0.785
	Posicional	0.280	0.151	0.187	0.422	0.815

Análogamente, se ha hecho este mismo estudio con el grafo de problemas: se ha realizado una comparativa de los resultados obtenidos con los diferentes métodos de agregación basán-

donos en el grafo de interacción de problemas. La Tabla 4.4 muestra estos resultados, donde se aprecian las diferencias con respecto a los resultados obtenidos con el grafo de usuarios. En este caso, no todos los valores obtenidos con los sistemas de votación son mejores que los obtenidos con el método de agregación basado en la similitud máxima. De hecho, para las métricas de similitud con pesos, excepto EW, el mejor valor obtenido es el del método de agregación basado en la similitud máxima. Para CN, AA, JN y EW los mejores resultados son los de la votación ponderada. Por métricas de evaluación, los mejores resultados para *Precision*, *Recall* y *F1* son lo que se consiguen con AA y el sistema de votación ponderada. Para MRR, lo mejores resultados se consiguen de nuevo con AA, pero con el método de agregación basado en la similitud máxima. Para *One hit*, sin embargo, el mejor resultado es el obtenido con WCN, WAA y WPA para el método de agregación basado en la similitud máxima (también para WPA con votación ponderada) coincidiendo el valor para las tres métricas de similitud.

Tabla 4.4: Resultados de la evaluación del sistema basado en el grafo de problemas: se muestran los resultados obtenidos con y sin sistemas de votación para $k = 5$ y $t = 20/10/2016$. Los mejores resultados obtenidos entre el grafo de usuarios y el de problemas está marcado en negrita. El mejor valor obtenido por métricas está marcados con *.

Métrica	Agregación	Precision	Recall	F1	MRR	1-hit
CN	Similitud	0.111	0.055	0.070	0.194	0.400
	Simple	0.114	0.047	0.064	0.165	0.354
	Ponderada	0.231	0.133	0.160	0.427	0.631
	Posicional	0.095	0.050	0.062	0.172	0.415
AA	Similitud	0.268	0.136	0.172	0.486*	0.677
	Simple	0.114	0.046	0.063	0.166	0.354
	Ponderada	0.295*	0.164*	0.201*	0.425	0.708
	Posicional	0.089	0.047	0.058	0.158	0.385
JN	Similitud	0.129	0.063	0.081	0.241	0.446
	Simple	0.114	0.047	0.064	0.165	0.354
	Ponderada	0.243	0.141	0.170	0.471	0.677
	Posicional	0.095	0.050	0.062	0.172	0.415
PA	Similitud	0.098	0.044	0.057	0.185	0.323
	Simple	0.114	0.047	0.064	0.165	0.354
	Ponderada	0.135	0.066	0.085	0.336	0.508
	Posicional	0.095	0.050	0.062	0.172	0.415
EW	Similitud	0.126	0.061	0.078	0.329	0.462
	Simple	0.114	0.047	0.064	0.165	0.354
	Ponderada	0.135	0.066	0.085	0.336	0.508
	Posicional	0.095	0.050	0.062	0.172	0.415
WCN	Similitud	0.252	0.132	0.165	0.421	0.785*
	Simple	0.114	0.047	0.064	0.165	0.354
	Ponderada	0.037	0.013	0.018	0.086	0.138
	Posicional	0.095	0.050	0.062	0.172	0.415
WAA	Similitud	0.249	0.131	0.164	0.423	0.785*
	Simple	0.114	0.047	0.064	0.165	0.354
	Ponderada	0.049	0.018	0.026	0.095	0.169
	Posicional	0.095	0.050	0.062	0.172	0.415
WPA	Similitud	0.265	0.137	0.172	0.428	0.785*
	Simple	0.114	0.047	0.064	0.165	0.354
	Ponderada	0.265	0.137	0.172	0.428	0.785*
	Posicional	0.095	0.050	0.062	0.172	0.415

4.6.3. Variaciones sobre el número de elementos recomendados

Durante nuestro análisis también se han generado gráficas para poder visualizar las diferencias de resultados entre los diferentes valores de k . En esta memoria sólo se van a analizar algunas de las gráficas generadas durante este trabajo como muestra del comportamiento general de los métodos propuestos. El resto de gráficas se encuentran disponibles en GitHub¹².

En este análisis vamos a hacer:

- Una comparación de los diferentes valores obtenidos para los distintos grafos cuando utilizamos el método de agregación basado en la similitud máxima.
- Una comparación de los diferentes valores obtenidos utilizando los sistemas de votación como métodos de agregación, sobre el grafo de usuarios.

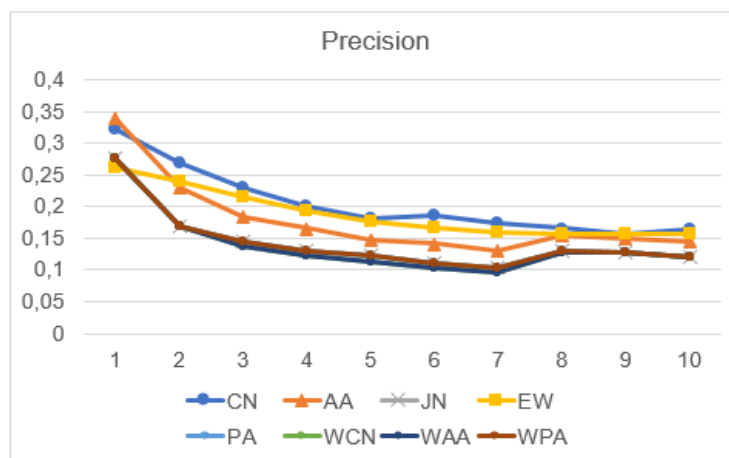


Figura 4.3: Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación *Precision* con el método de agregación basado en la similitud máxima

¹²<https://github.com/martcaro/TrabajoFinDeMaster>

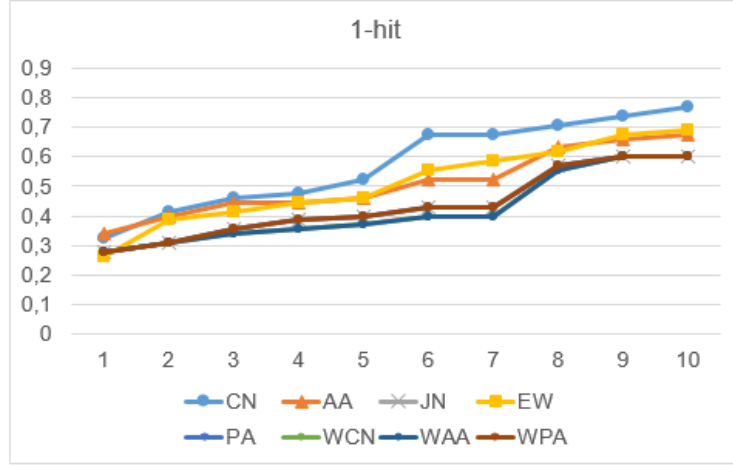


Figura 4.4: Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación *One hit* con el método de agregación basado en la similitud máxima

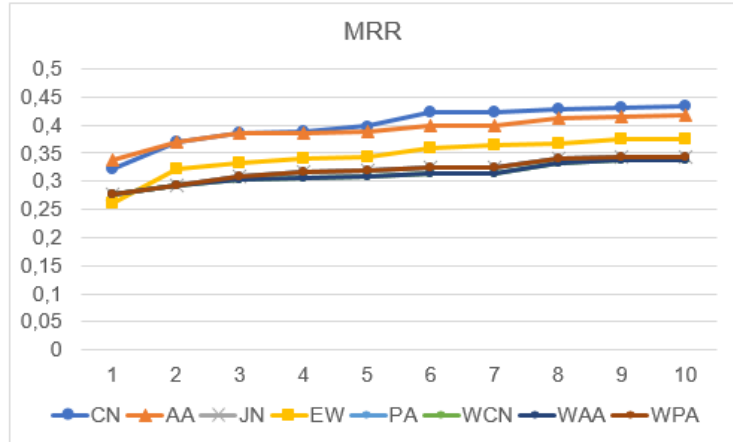


Figura 4.5: Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación MRR con el método de agregación basado en la similitud máxima

En primer lugar, se hizo una comparativa de los resultados obtenidos con el grafo de usuarios y el método de agregación basado en la similitud máxima. Teniendo esto en cuenta, en la Figura 4.3, se observan los distintos valores que toma k usando la métrica de evaluación *Precision*. Se puede observar que el valor de *Precision* se reduce poco a poco conforme va aumentando k . En la Figura 4.4 observamos de nuevo los valores que toma k utilizando la métrica de evaluación *One hit*. El valor de k aumenta en este caso, ya que,

cuanto más problemas se recomiendan, más probabilidades hay de que alguno de los problemas recomendados se encuentre en el conjunto de datos de evaluación para ese usuario. Por último, en la Figura 4.5, se encuentran los valores de k para la métrica de evaluación 4.5. El valor de MRR se incrementa también poco a poco asintóticamente hasta llegar a un valor constante a la vez que aumenta k .

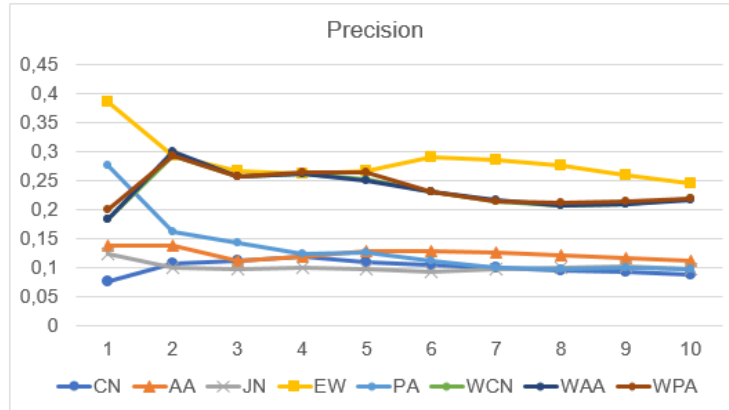


Figura 4.6: Valores obtenidos para el grafo de problemas, $t = 20/10/2016$ y la métrica de evaluación *Precision* con el método de agregación basado en la similitud máxima

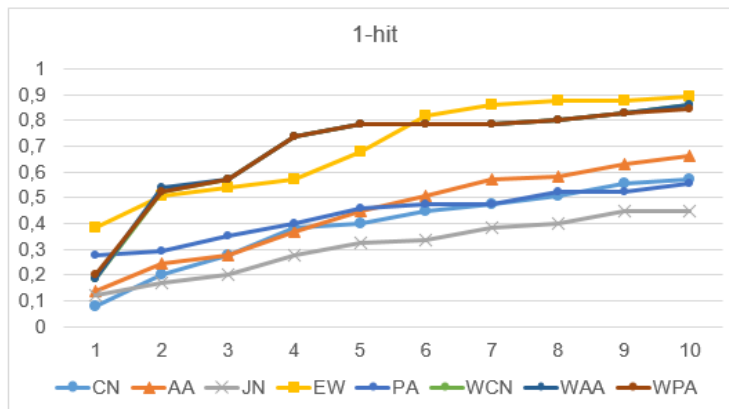


Figura 4.7: Valores obtenidos para el grafo de problemas, $t = 20/10/2016$ y la métrica de evaluación *One hit* con el método de agregación basado en la similitud máxima

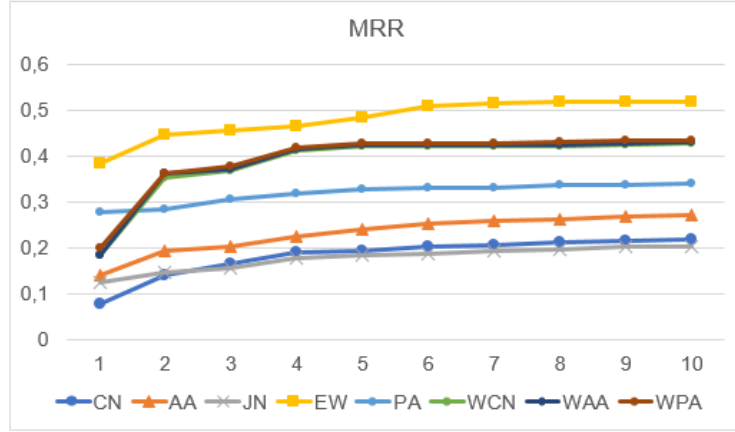


Figura 4.8: Valores obtenidos para el grafo de problemas, $t = 20/10/2016$ y la métrica de evaluación MRR con el método de agregación basado en la similitud máxima

En el caso del grafo de problemas (en las Figuras 4.6, 4.7 y 4.8), se distinguen a simple vista dos zonas claramente diferenciadas: los resultados más altos se concentran en las métricas de similitud con pesos, mientras que los bajos son los que se obtienen con las métricas sin pesos. Se hace patente sobre todo con las métricas de evaluación *One hit* (Figura 4.7) y *Precision* (Figura 4.6), aunque con MRR (Figura 4.8) también se muestra la superioridad de los valores de las métricas con pesos. De nuevo, con *Precision* el valor obtenido decrece cuando crece el valor de k , aumenta asintóticamente con *MRR* y crece cuando se usa *One hit*.

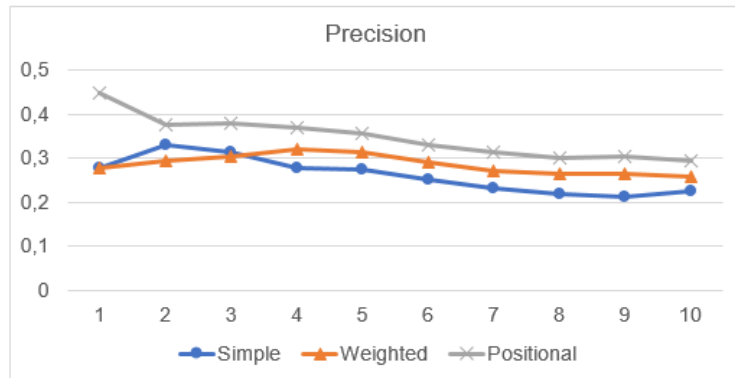


Figura 4.9: Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación *Precision* con los sistemas de votación como método de agregación y CN como métrica de similitud

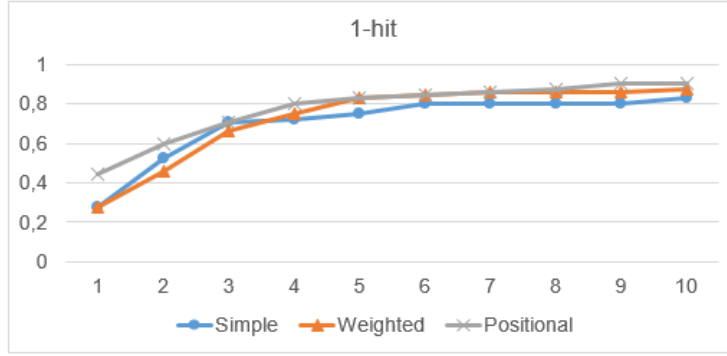


Figura 4.10: Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación *One hit* con los sistemas de votación como método de agregación y CN como métrica de similitud

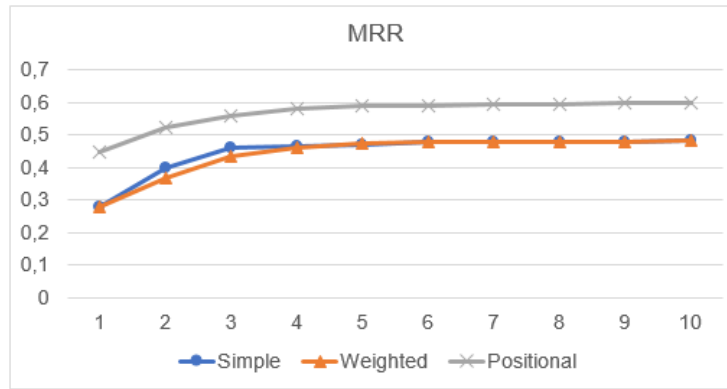


Figura 4.11: Valores obtenidos para el grafo de usuarios, $t = 20/10/2016$ y la métrica de evaluación MRR con los sistemas de votación como método de agregación y CN como métrica de similitud

El análisis de la evolución de los resultados utilizando sistemas de votación sobre el grafo de usuarios se muestra en las Figuras 4.9, 4.10 y 4.11. Son los resultados obtenidos con $t = 20/10/2016$ aplicando la métrica de similitud CN (que es la métrica con la que se han obtenido los mejores resultados con el grafo de usuarios). Con MRR, los valores crecen asintóticamente a un valor constante a la vez que aumenta el valor de k , mientras que el valor de *Precision* decrece al crecer k y el valor de *One hit* aumenta al crecer k .

Como hemos visto, la evolución de k tiene siempre la misma tendencia, independien-

temente del grafo sobre el que se basen las recomendaciones, los métodos de agregación utilizados, las métricas de similitud usadas y las métricas de evaluación tenidas en cuenta en nuestro análisis. La tendencia de crecimiento con respecto a k de *One hit* y MRR ocurre porque cuanto mayor sea el número de elementos a recomendar más fácil será que alguno de los problemas recomendados le pueda interesar al usuario. En cuanto a la tendencia de descenso de *Precision* ocurre porque la valoración de los aciertos se tiene en cuenta con respecto a todos los productos recomendados. Aunque es más fácil acertar con un valor de k mayor, también la precisión de las recomendaciones es menor al tener más elementos seleccionados o recomendados.

4.7. Conclusiones

El proceso de evaluación de un sistema recomendador es imprescindible para determinar si el sistema propuesto funciona según lo esperado y cumple lo requerido para cubrir las necesidades de los usuarios de la plataforma para la que se propone el recomendador. En nuestros experimentos se muestra la obligación de pasar por todas las etapas de la evaluación de un recomendador. Tras el trabajo realizado en este Trabajo Fin de Máster, se pueden concluir varias ideas en base a la gran cantidad de datos recopilados gracias al proceso de evaluación realizado en nuestros experimentos. Las ideas más importantes con las que podemos concluir el análisis de resultados son:

- Como hemos visto a lo largo del capítulo, es importante experimentar con las diferentes variables del proceso de evaluación para determinar cómo adaptar el sistema recomendador de la mejor manera posible al conjunto de datos sobre el que queremos aplicarlo: grafos de interacción, métricas de similitud, número de problemas a recomendar y métricas de evaluación.
- De forma global, los mejores resultados conseguidos son los obtenidos por los métodos

de recomendación basados en el grafo de usuarios que usan los sistemas de votación.

- En general, el grafo de problemas es más prometedor si se utilizan métricas sin pesos y con el método de agregación basado en la similitud máxima.
- La utilización de sistemas de votación puede dar lugar a los mejores resultados globales si los aplicamos sobre el grafo de usuarios, y a los peores si lo aplicamos sobre el grafo de problemas.
- Decidir qué valor de k elegir depende de cuáles sean nuestras necesidades. En nuestra evaluación, obtenemos los mejores resultados cuanto mayor es k : es más probable acertar cuantas más recomendaciones se hagan al usuario. Sin embargo, se tiene que elegir un valor de k que nos aporte los mejores resultados posibles con el menor número de problemas recomendados. Esto hará que se proporcione al usuario problemas que puedan interesarle pero sin saturarle.
- También es importante elegir métricas de evaluación que nos puedan dar información sobre diferentes aspectos del sistema.

Tras haber realizado la evaluación del recomendador y el análisis de los resultados obtenidos, en el siguiente capítulo se va a proceder a comentar conclusiones sobre el trabajo realizado en este proyecto y a determinar cuáles serían las líneas de trabajo futuro.

Capítulo 5

Conclusiones

El desarrollo de las nuevas tecnologías ha impulsado el crecimiento y expansión del consumo a través de Internet. La oferta de páginas web y aplicaciones con las que se pueden adquirir productos es inmensa y, por tanto, también la cantidad de productos que nos ofrece Internet. Los usuarios pueden encontrar entonces lo que necesitan entre tanta variedad, pero esto también provoca que la búsqueda resulte en muchas ocasiones complicada. De este problema, surgen los sistemas de recomendación. Son sistemas que ayudan a los usuarios en la búsqueda de productos que puedan interesarles, proponiéndoles diferentes productos en base a sus gustos. Sin embargo, aunque hay muchas plataformas donde ya se han implantado, también hay muchas otras donde no existen sistemas de recomendación pero en las cuáles sería interesante que se implantasen. Una de estas plataformas son los jueces en línea. Son plataformas que ofrecen una amplia oferta de problemas de programación y estructuras de datos y algoritmos. Son populares por su uso como jueces en concursos de programación y también por la posibilidad que tienen estudiantes de informática y ciencias de la computación de usarlos como herramienta para practicar ejercicios. Los usuarios pueden subir a la plataforma la solución a un problema determinado y el juez le proporciona un veredicto para la solución entregada. La gran cantidad de problemas que pueden contener estos jueces suele ser enorme, de ahí la necesidad de orientar al usuario en la tarea de seleccionar aquellos ejercicios que más puedan interesarle.

El razonamiento anterior es la motivación principal de este Trabajo Fin de Máster, que ha consistido en un trabajo de investigación en el que se han propuesto y evaluado métodos de recomendación basados en grafos de interacción para jueces en línea.

El comienzo del trabajo es siempre una etapa básica de descubrimiento de información y adquisición de nuevos conocimientos. En este caso, conocer toda la información relativa a los sistemas de recomendación era imprescindible para empezar este Trabajo Fin de Máster. Fue la forma de aprender en qué consistían los sistemas de recomendación, las diferentes formas de generar recomendaciones, las aplicaciones que podían tener, la manera en que se pueden evaluar (etapa muy importante en nuestro trabajo) y la importancia actual de su existencia.

Tras estudiar en qué consisten los sistemas de recomendación, era importante también hacer un análisis de las carencias de los jueces en línea en materia de orientación a usuarios, debido a que estos eran los sistemas a los que íbamos a aplicar nuestros recomendadores. Como se ha visto, existe una necesidad real de incluir sistemas de recomendación en jueces en línea, ya que puede ser altamente beneficioso para los estudiantes.

Después de hacer un análisis de las bases de los que va a ser el Trabajo Fin de Máster, el siguiente paso fue hacer una réplica de lo realizado en el artículo de investigación *"Similarity Metrics from Social Network Analysis for Content Recommender Systems"* [21], presentado en el congreso ICCBR *International Conference on Case-Based Reasoning* de 2016. En este artículo se hacían unas primeras propuestas de métodos de recomendación para el juez en línea Acepta el Reto. El objetivo de estudiar este trabajo era familiarizarse con las principales técnicas a utilizar para después hacer una expansión, incluyendo otras muchas nuevas propuestas. Tras el trabajo realizado, se puede concluir que este ejercicio fue la mejor forma de comenzar el Trabajo Fin de Máster, ya que sentó las bases del resto de la investigación

y aportó muchos conocimientos nuevos necesarios para llevar a cabo el trabajo. A partir de esto, se realizaron muchas nuevas propuestas e implementaciones que ampliaron en gran medida la investigación publicada en el 2016.

Para la generación de nuestros métodos de recomendación se han aplicado diferentes técnicas. La base de las recomendaciones se hace en función de grafos de interacción que representan las interacciones llevadas a cabo en un juez en línea. Se han construido dos grafos de interacción distintos (de problemas y de usuarios) con los que se generan las recomendaciones de forma diferente. Sobre estos grafos se han aplicado técnicas de predicción de enlaces, otro pilar importante en nuestros métodos de recomendación, ya que, con ellas, hemos podido obtener información de similitud imprescindible para poder generar las recomendaciones. Se han propuesto también métodos de agregación con el objetivo de hacer recomendaciones efectivas teniendo en cuenta las similitudes entre los elementos que interactúan en el juez. Como se observaría después, la utilización de los diferentes grafos de interacción, métricas de similitud y sistemas de agregación puede variar significativamente el grado de efectividad del sistema recomendador.

Tras proponer nuestros métodos de recomendación, había que realizar la evaluación de los mismos. En total, se realizaron una amplia cantidad de experimentos teniendo en cuenta que se implementaron los 64 métodos de recomendación propuestos con diferentes parámetros: los diferentes tamaños de listas de recomendación, las diferentes métricas de similitud usadas y las diferentes métricas de evaluación utilizadas. Todos estos experimentos han dado lugar a gran cantidad de datos, sin los cuáles no hubiéramos podido hacer el análisis de las propuestas realizadas. Por otro lado, aunque esto tiene beneficios a nivel investigador de nuestro trabajo, también supuso un esfuerzo de análisis debido a la gran diversidad de datos y la dificultad de encontrar patrones en tal variedad de cifras. Como conclusiones del análisis realizado se puede destacar que los mejores resultados obtenidos son los conseguidos con el

sistema de recomendación basado en grafos de interacción de usuarios utilizando además sistemas de votación. La elección del tamaño de la lista de recomendación puede cambiar los resultados obtenidos: es mejor encontrar un valor medio donde el resultado sea aceptable que una lista enorme de recomendaciones que pueda saturar al usuario.

Este Trabajo Fin de Máster ha tenido buena acogida a nivel investigador generando la publicación *"Similar Users or Similar Items? Comparing Similarity-based Approaches for Recommender Systems in Online Judges"* [10] en el congreso ICCBR *International Conference on Case-Based Reasoning* de 2017, donde se recoge buena parte de lo presentado en esta memoria.

Finalmente, concluir que este Trabajo de Fin de Máster ha sido muy satisfactorio por la cantidad de conocimientos aprendidos y de resultados obtenidos y también como forma de introducción a la investigación para realizar una tesis doctoral.

5.1. Conclusions

The offer of web pages and applications with which you can get products is immense and, therefore, also the quantity of products offered by the Internet. Then, users can find what they need, but the search can be complicated. From this problem, recommender systems appear. They are systems that help the users in the search of products that may interest them. However, although there are many platforms where they have already been implemented, there are many platforms where it would be interesting to introduce recommender systems. One of these platforms are online judges. They are platforms that offer a wide range of programming problems. They are popular for their use as judges in programming contests and also for their use as a tool for practicing exercises for computer science students. Users can upload the solution to a given problem and the judge provides a verdict for this solu-

tion. The great amount of problems that these judges can contain is usually huge, hence the need to guide the user in the task of selecting those exercises that may be interesting to him.

The previous reasoning is the main motivation of this Final Master's Work, which has consisted in a research work where we proposed recommender methods based on interaction graphs for online judges.

The beginning of work is always a basic stage for discovering information and acquiring new knowledge. In this case, knowing all the information related to the recommender systems was essential to begin this Final Master's Work. It was the way to learn what the recommender systems are, the different ways of generating recommendations, their applications, the way they can be evaluated (very important stage in our work) and the current importance of their existence.

After examining what the recommendation systems consist in, it was also important to make an analysis of online judges' lacks, because these were the systems to which our recommendations would apply. As we have seen, there is a real need to include recommender systems in online judges. It can be highly beneficial to students.

After that, the next step was to replicate what was done in the research article *"Similarity Metrics from Social Network Analysis for Content Recommender Systems"* [21], presented at the International Conference on Case-Based Reasoning (ICCBR) in 2016. This article made a few recommender systems proposals for the online judge Acepta el Reto (Take On the Challenge). The objective of studying this work was to familiarize ourselves with the main techniques which we would use in this work. After the work done, it can be concluded that this exercise was the best way to begin the Master's End Work. It was the basis of the rest of the research and provided many new knowledge necessary to carry out the work.

From this, many new proposals and implementations were made.

Different techniques have been applied for the generation of our recommendation methods. Our recommender systems work based on interaction graphs that represent the interactions carried out in an online judge. Two different interaction graphs (problems graph and users graph) have been built. On these graphs have been applied link prediction techniques, another important pillar in our recommender methods, since with them, we have been able to obtain essential similarity information in order to generate the recommendations. Aggregation methods have also been proposed with the aim of making effective recommendations taking into account the similarities between the elements that interact in the judge. The use of different interaction graphs, similarity metrics and aggregation systems can significantly vary the degree of effectiveness of the recommender system.

After proposing our recommender methods, they had to be evaluated. In total, a large number of experiments were carried out taking into account that the 64 recommender methods proposed with different parameters were implemented: the different sizes of recommendation lists, the different similarity metrics used and the different evaluation metrics used. All these experiments have produced a large amount of data. Analyzing the data was a effort due to the great diversity of data and the difficulty of finding patterns in such a variety of numbers. As conclusions of the analysis performed, we can say that the best results obtained are those obtained with the recommender systems based on users interaction graphs also using voting systems. Choosing the size of the recommendation list can change the results obtained: it is better to find an average value where the result is acceptable than a huge list of recommendations that can saturate the user.

This Final Master's Work has been well received as a research work. It was accepted the research paper *"Similar Users or Similar Items? Comparing Similarity-Based Approaches*

for Recommender Systems in Online Judges" [10] at International Conference on Case-Based Reasoning (ICCBR) of 2017.

Finally, we conclude that this Master's Degree Project has been very satisfactory due to the amount of knowledge learned and results obtained and also as a way of introduction to carry out a doctoral thesis.

5.2. Líneas de trabajo futuro

Una de las líneas de futuro inmediata a este trabajo sería evaluar los métodos de recomendación propuestos sobre otros jueces en línea, adaptando los métodos a cada juez. Sería adecuado realizar esto para corroborar que los sistemas de recomendación propuestos se pueden usar también en otros jueces en línea, además de en ACR. Especialmente interesante sería hacer la evaluación con conjuntos de datos muchos más grandes que los que tenemos en ACR, por ejemplo, el conjunto de datos del juez en línea de la UVa. Con ello, evaluaríamos la escalabilidad de los procesos propuestos.

Otra línea de futuro a llevar a cabo sería la de hacer una evaluación de nuestros métodos de recomendación con usuarios reales. Las evaluaciones con usuarios son más fiables que las evaluaciones *offline*, por lo que previsiblemente obtendríamos nuevos datos interesantes y útiles para determinar la validez de nuestros modelos.

Teniendo en cuenta que el objetivo de los usuarios de los jueces en línea es aprender a programar y a utilizar estructuras de datos y algoritmos, se puede contemplar como línea de trabajo futuro la posibilidad de utilizar los intereses pedagógicos de los usuarios como factor para hacer las recomendaciones. De esta forma, se podría evaluar también la evolución del aprendizaje de los usuarios gracias a los métodos de recomendación propuestos.

Por otro lado, también se podrían utilizar otras técnicas en nuestros modelos de recomendación. Por ejemplo, utilizar algunas de las técnicas de predicción de enlaces vistas en esta memoria, como las técnicas basadas en aprendizaje, o utilizar alguna otra de las métricas de evaluación existentes que no se han utilizado en este trabajo.

Sin embargo, la línea de trabajo futuro más prometedora sería la realización de un *framework* que pueda aplicarse de forma general a cualquier plataforma, tanto jueces en línea como a otros tipos de sistemas. La idea básica para generalizar nuestros sistemas de recomendación es determinar cuáles son los procedimientos en común que utilizan y los puntos de variabilidad a adaptar para cada situación: la base de datos, el tipo de grafo de interacciones a generar (de usuarios o productos), qué métricas de similitud utilizar, el sistema de agregación a utilizar o el tamaño de la lista de recomendaciones que se quiere obtener. La generación de este *framework* puede facilitar la incorporación de sistemas de recomendación a plataformas donde los sistemas de recomendación basados en filtrado colaborativo, los basados en conocimiento y los basados en contenido no se pueden aplicar.

Bibliografía

- [1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.
- [2] Charu C Aggarwal. *Recommender Systems*. Springer, 2016.
- [3] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.
- [4] Albert-László Barabási. Network science book. *Boston, MA: Center for Complex Network, Northeastern University. Available online at: <http://barabasi.com/networksciencebook>*, 2014.
- [5] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.
- [6] Alejandro Bellogin, Pablo Castells, and Ivan Cantador. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 333–336. ACM, 2011.
- [7] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.
- [8] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

- [9] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.
- [10] Marta Caro-Martinez and Guillermo Jimenez-Diaz. *Similar Users or Similar Items? Comparing Similarity-based Approaches for Recommender Systems in Online Judges*. En prensa, 2017.
- [11] L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in physics*, 56(1):167–242, 2007.
- [12] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer, 2011.
- [13] Simon Doods, Alejandro Bellogín, Toon De Pessemier, and Luc Martens. A framework for dataset benchmarking and its application to a new movie rating dataset. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):41, 2016.
- [14] Sergey N Dorogovtsev and Jose FF Mendes. Evolution of networks. *Advances in physics*, 51(4):1079–1187, 2002.
- [15] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):10, 2011.
- [16] Alexander Felfernig and Robin Burke. Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th International Conference on Electronic Commerce*, page 3. ACM, 2008.
- [17] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evalua-

- ting collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.
- [18] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115, 2004.
 - [19] Zan Huang, Wingyan Chung, and Hsinchun Chen. A graph model for e-commerce recommender systems. *Journal of the Association for Information Science and Technology*, 55(3):259–274, 2004.
 - [20] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.
 - [21] Guillermo Jiménez-Díaz, Pedro Pablo Gómez Martín, Marco Antonio Gómez-Martín, and Antonio A. Sánchez-Ruiz. *Similarity Metrics from Social Network Analysis for Content Recommender Systems*, volume 9969 of *LNCS*, pages 203–217. Springer, 2016.
 - [22] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 145–186. Springer, 2011.
 - [23] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.
 - [24] Adrian Kosowski, Michał Małafiejski, and Tomasz Noiński. Application of an online judge & tester system in academic tuition. *Advances in Web Based Learning-ICWL 2007*, pages 343–354, 2008.
 - [25] Xiaoyi Li, Nan Du, Hui Li, Kang Li, Jing Gao, and Aidong Zhang. A deep learning approach to link prediction in dynamic networks. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 289–297. SIAM, 2014.
 - [26] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks.

- In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 556–559. ACM, 2003.
- [27] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.
 - [28] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.
 - [29] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.
 - [30] Jéssica Martín Jabón, Javier Martín Moreno-Manzanaro, Pablo Suárez Díaz, Pedro Pablo Gómez Martín, and Marco Antonio Gómez-Martín. Juez automático para la evaluación de problemas de programación en los primeros cursos de estudios de informática. Disponible en E-Prints Complutense. 2013.
 - [31] Lorraine McGinty and Barry Smyth. On the role of diversity in conversational recommender systems. pages 276–290, 2003.
 - [32] Lorraine McGinty and Barry Smyth. Improving the performance of recommender systems that use critiquing. pages 114–132, 2005.
 - [33] Sean M McNee, John Riedl, and Joseph A Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101. ACM, 2006.
 - [34] Aditya Menon and Charles Elkan. Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases*, pages 437–452, 2011.

- [35] Joshua O'Madadhain, Jon Hutchins, and Padhraic Smyth. Prediction and ranking algorithms for event-based network data. *ACM SIGKDD Explorations Newsletter*, 7(2):23–30, 2005.
- [36] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059–10072, 2012.
- [37] Michael Pazzani and Daniel Billsus. Content-based recommendation systems. *The adaptive web*, pages 325–341, Springer, 2007.
- [38] Daniele Quercia, Mansoureh Bodaghi, and Jon Crowcroft. Loosing friends on facebook. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 251–254. ACM, 2012.
- [39] Lara Quijano Sánchez. *Impacto de los factores u organizaciones sociales en los procesos de recomendación para grupos*. PhD thesis, Universidad Complutense de Madrid, 2015.
- [40] Matías Ricotta Elizalde and Esteban Santamarina. Predicción de enlaces en redes sociales modeladas por grafos. 2017.
- [41] Almudena Ruiz Iniesta. *Estrategias de recomendación basadas en conocimiento para la localización personalizada de recursos en repositorios educativos*. PhD thesis, Universidad Complutense de Madrid, 2014.
- [42] Alan Said, Alejandro Bellogín, and Arjen De Vries. A top-n recommender system evaluation protocol inspired by deployed systems. In *ACM RecSys 2013 Workshop on Large-Scale Recommender Systems (LSRS'13), Hong Kong*, 2013.
- [43] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

- [44] Purnamrita Sarkar, Deepayan Chakrabarti, and Michael Jordan. Nonparametric link prediction in dynamic networks. *arXiv preprint arXiv:1206.6394*, 2012.
- [45] JHJB Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. *The adaptive web*, pages 291–324, Springer, 2007.
- [46] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.
- [47] Hideo Shimazu. Expertclerk: navigating shoppers’ buying process with the combination of asking and proposing. In *Proceedings of the 17th international joint conference on Artificial intelligence-Volume 2*, pages 1443–1448. Morgan Kaufmann Publishers Inc., 2001.
- [48] Barry Smyth. Case-based recommendation. *The adaptive web*, pages 342–376, 2007.
- [49] Carlos Ferro Soto, Ana Isabel Martínez Senra, and M^a Carmen Otero Neira. Ventajas del uso de las TICs en el proceso de enseñanza-aprendizaje desde la óptica de los docentes universitarios españoles. *EDUTECH. Revista electrónica de Tecnología educativa*, (29), 2009.
- [50] Lei Tang and Huan Liu. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–137, 2010.
- [51] Jorge Valverde-Rebaza and Alneu de Andrade Lopes. Exploiting behaviors of communities of twitter users for link prediction. *Social Network Analysis and Mining*, 3(4):1063–1074, 2013.
- [52] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the*

29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 501–508. ACM, 2006.

- [53] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. Unified relevance models for rating prediction in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 26(3):16, 2008.
- [54] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, 58(1):1–38, 2015.
- [55] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. *Social media mining: an introduction*. Cambridge University Press, 2014.
- [56] Markus Zanker, Markus Jessenitschnig, and Wolfgang Schmid. Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15(4):574–595, 2010.
- [57] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. *Proceedings of the tenth international conference on Information and knowledge management*. pages 25–32, 2001.
- [58] Tao Zhou, Jie Ren, Matúš Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Phys. Rev. E*, 76:046115, Oct 2007.

